

АҚПАРАТТЫҚ ТЕХНОЛОГИЯЛАР

ЗАМАНАУИ САНДЫҚ ӘЛЕМГЕ КІРІСПЕ

РИЧАРД ФОКС



CRC Press
Taylor & Francis Group

A CHAPMAN & HALL BOOK

6000 Broken Sound Parkway NW, Suite 300 Бока-Ратон, FL 33487-2742

© 2013 Тэйлор&Фрэнсис Групп , ОсОО

CRC Press Taylor & Francis Group, АҚШ үкіметінің түпнұсқалық жұмысына бизнес ақпараты ұмтылмайды

Баспа Америка Құрама Штаттарында қышқылсыз қағазда басып шығарылды. Нобайлану күні : 20121023

Кітаптың халықаралық стандартты нөмірі : 978-1-4665-6828-0 (жұмсақ мұқаба)

Бұл кітаптың құрамында сенімді және беделді кітаптардан алынған ақпарат бар. Сенімді деректер мен ақпаратты жариялау үшін үлкен әрекет етті, бірақ автор мен баспагер өзіне барлық материалдарды немесе салдарды пайдалануды жауапкершілігіне ала алмайды. Авторлар мен баспагерлер барлық материалдардың авторлық құқық иелерін бақылауға тырысты, егер осындай түрде жариялауға рұқсат алынған жоқ болса, осы жарияланымдарды қайтадан өндіруді және авторлық құқық иелерінің алдында кешірім сұрау керек. Егер қандай да бір материалдың авторлық құқығы расталмаған болса, болашақта кез келген түзетуге қайта басып шығару үшін бізге хабарлауыңызды өтінеміз. Бұл жағдайларды қоспағанда, рұқсат етілген заңға сәйкес, авторлық құқық туралы АҚШ-тың бірде-бір бөлігінде бұл кітапты қайта басу мүмкін емес, оны жаңадан өндіруге беріліп немесе кез келген нысанда және кез келген электрондық, механикалық немесе өзге де құралдармен пайдалануға, қазір белгілі немесе келешекте өнертабыстар, оның ішінде фотокөшірме, микрофильмдеу, жазба немесе қандай да бір ақпаратты сақтау және іздеу жүйесі, баспагерлер тарапынан жазбаша рұқсатынсыз әрекет етуге болмайды.

Материалдарды электрондық түрде көшіру немесе пайдалану үшін рұқсат алу, яғни www.copyright.com (HTTP: //www.copyright.com/) сайтына кіріп немесе Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400 байланысқа шығу қажет. ККК коммерциялық емес болып келеді, ол лицензияны тіркеуді және пайдаланушылардың кең ауқымы үшін ұсынады. КТС фотокөшірмелі ұйымдар үшін лицензия ұсынылған, жеке төлеу жүйесі анықталған.

Хабарлама туралы тауар таңбалары: өнім немесе фирмалық сауда белгілерінің атауы болуы мүмкін немесе тіркелген сауда белгілері және сәйкестендіру мақсатында және ниетінсіз қандай да бір қағидаларды бұзуға түсініктемелер ғана пайдаланылады.

Конгресс кітапханасында берілген жариялымдар

Фокс, Ричард, 1964-

Ақпараттық технологиялар: заманауи сандық әлеміне кіріспе / Ричард Фокс.

Библиографиялық сілтемелер мен индекс қосылған. ISBN 978-1-4665-6828-0 (Жұмсақ мұқаба : Сілтілік қағаз)

1. Ақпараттық технологиялар . I. Атауы.

T58.5.F695 2013

004 - DC23 2012036916

Taylor & Francis <http://www.taylorandfrancis.com> және CRC Press <http://www.crcpress.com> веб-сайтына кіріңіз .

Мазмұны

1 Тарау. Ақпараттық технологияларға кіріспе	6
2 Тарау. Компьютерлерді ұйымдастыру және жинақтамалар	31
3 Тарау. Есептеудің екілік жүйесі	70
4 Тарау. Операциялық жүйе қағидасына кіріспе	110
5 Тарау. Файлдар, Тізімдемелер және файлдық жүйе	149
8 Тарау. Операциялық жүйелердің тарихы	200
9 Тарау. Bash қабықшасы және түзету	221
10 Тарау. Тұрақты өрнек	261
11 Тарау. Үдерістер мен қызметтер	290
12 Тарау. Үдерістер мен қызметтер	333
13 Тарау. Бағдарламалық жасақтама	383
14 Тарау. Бағдарламалау	406
15 Тарау. Ақпарат	448
16 Тарау. Ақпараттық технология саласындағы мансап	495

1. Ақпараттық технологияларға кіріспе

Оқу құралында ақпараттық технологиялар (АТ) саласы кіріспені білдіреді, АТ - аралас салаларда студенттер үшін арналған. Бұл тарауда әр түрлі АТ адамның мансаптық рөлін қарастырады, жүйелік әкімшілендіру, сондай-ақ түрлері АТ талап етілетін дағдыларын сипаттайды. Бұл тарауда құрайтын элементтер, компьютерлер, бағдарламалық қамтамасыз ету, және пайдаланушылар инфрақұрылымды көрсетті.

Осы тараудың оқыту мақсаты келесідей болып табылады

- АТ-мансапты баяндау мен түрлерін ажырату.
- АТ сипаттау жинағы дағдыларын, қажетті табысқа жету үшін.
- Жабдық түрлерін тауып, компьютерлік жүйесіне енгізу.
- Ажырату компоненттері арасындағы компьютерлік жүйесін аппараттық құралдарын сипаттауы, бағдарламалық және пайдаланушыларды қамтамасыз ету.

Ақпараттық технология дегеніміз не?

Сонымен, бұл ақпараттық технологиялар (АТ) кез келген жағдайда не болып табылады? Бұл термин бірнеше міндетті заттарды сипаттау үшін, деректерді жинау және өңдеу, олардың ақпаратты тарату мүмкіндігі технологияларды пайдалана отырып, жиынтығы бар адамдар, АТ-инфрақұрылымды ұстау үшін (компьютерлер, желілер, операциялық жүйе) өзінің міндеттерін шешуге ақпаратты технология мүмкіндік береді. Жалпы, біздің ойымызша, бұл технологияны құру, сақтау және ақпараттың қолжетімділігі пайдаланылады. Басқа сөздермен айтқанда, ол адамдардың есептеуіш ресурстарын біріктіреді, бағдарламалық деректер мен компьютерлік желілерді қамтамасыз етеді.

АТ-қызметкер, кейде жалпы атаумен "АТ" адамдары деп аталады, АТ-ны жеткізу және қолдау, олардың жұмысы. Оларға жататындарды компьютерлік инженерлер жобалайды және жасайды.

Компьютерлік чиптер, компьютерлік ғалымдар, компьютерлер мен әкімшілер үшін АТ-инфрақұрылымды ұйым бағдарламалық қамтамасыз етеді.

АТ облысында сіздің рөліңіз қандай болады? АТ қызметкерлерінің көптеген әртүрлі міндеттері бар. Кейбір жағдайларда барлығы үшін бір адам ғана болуы мүмкін, бірақ көптеген бірнеше жағдайларда қажет болуы мүмкін, бәлкім ондаған немесе жүздеген жобасына қатысушы қызметкерлері, олардың әрқайсысы жекелеп алмасады. Көпшілігі АТ қызметкерлері, алайда, екі жалпы функциялары бар: басқару және қолдау. АТ-инфрақұрылымының әкімшісі қандай да бір бөлігі болып табылады. Әкімшінің кейбір рөлдерін 1.1 кестеде толық қарастырайық. АТ-да кең таралған рөл жүйелік әкімшілік болып табылады. Жиелік әкімшілік бұл процесс операциялық жүйелерді қолдау компьютерлік жүйелер. Оқшауланған компьютерде жүйелік әкімшілік минималды болып табылады және әдетте, компьютердің көмегімен адамдарға қалдырады. Дегенмен, компьютерлер желісі немесе компьютерлер үшін, файлдарды немесе басқа да ресурстарды бірлесіп пайдаланады, жүйелік әкімшілік маңыздырақ және күрделі міндет болып келеді. Жүйенің әкімшісі жүйені орындайтын адам (немесе адамдар) болып табылады. Компьютерлік жүйелерді (компьютерлер, желі жүйе) ұстау өзіне түсіну бағдарламалықты, аппараттық қамтамасыз етуді және бағдарламалауды қамтиды. Бағдарламалық қамтамасыз ету тұрғысынан, басқаруды орнатуды талап етеді, ол қол жетімді қылады, мәселелерді жою туындаған кезінде пайдалану және бұл бағдарламалық қамтамасыз ету тиімді жұмыс істейді. Бұдан басқа, әкімшісі (лар) операциялық жүйені жақсы түсіну керек, бағдарламалықты орнатуды қамтамасыз ету тиісті, осы ұйым үшін тіркелуін жасау және сырттан шабуыл жүйесіне шаралар қабылданады.

Аппараттық тұрғысынан әкімшілік жаңа жабдықтарды орнатуды және қолданыстағы жабдықтар ақауларын жоюды талап етеді. Бұл мүмкін немесе мүмкін емес міндеттерінің деңгейінің қамтуы, төмендігі компоненттерін жөндеу сияқты және желілік кабельді тарату, сондай-ақ бағдарламалық орнатуды қамтамасыз етудің драйвері құрылғысы әр кезде қосылып жаңа жабдықтар қажет болуы мүмкін.

Бағдарламалау тұрғысынан, операциялық жүйелер "жіңішке күйге келтіру" және осылайша, әкімшілер өз сценарийлерін жиі жазу, қарапайым қабығының орындау үшін талап етеді.

1.1-кесте АТ-дағы әкімшіліктің рөлі

Рөлі	Жұмысы\ міндеттер
Жүйелік әкімшілік	Басқару компьютерлерін ұйымдастыру; бағдарламалық қамтамасыз етуді орнату; өзгерту / операциялық жүйені жаңарту; тіркелгісін жасау; жөндіктіру; пайдаланушылардың жөндіктіруі; ақаулықтарды жою жүйесі; қосу жабдығы
Желі әкімшілігі	Сатып алу, орнату және компьютерлік желілерді іске қосу; қолдау, компьютерлік желі; желіде ақаулықтарды жою; қорғалған желісі басып кіруді қондыру
Дерекқор әкімшілігі	орнату және қолдау, деректер базасын басқару жүйесі және деректер базасы; мәліметтер базасын резервтік көшіру; тіркелгісін жасау; пайдаланушыларды үйрету
Web әкімшісі	Орнату, баптау және веб-сервер арқылы веб сайтты басқару, қорғай, сайтты дамытушылармен жұмыс жасау
Веб – әзірлеушілер	веб-әзірлеушілердің дизайны және веб-беттерді жасау мен веб-беттер үшін скриптілер; веб-сайттарды шығарып салу және веб-сервер арқылы веб-сайтты орнату, күйге келтіру; қауіпсіз веб-сайт; жұмыс әзірлеушілер.
Әкімшілердің қауіпсіздік орнату	баптау және брандмауэрды

басқаруы; қауіпсіздік саясатын құруы; компьютерлік жүйенің ақауларын жоюы (соның ішінде желі)

Linuxта, мысалы көптеген компоненттер операциялық жүйесі конфигурациялық файлға сүйенеді. Олар жиі сценарийді өзгертеді. Әкімші болуы конфигурациялық файлды сәйкестендіруге және оны бейімдеп редакциялауға мүмкіндігі бар, бұл компонент ұйым аясында жұмыс істейді. Скриптерді жазу мақсаты автоматтандыру процестерінен тұрады, сондықтан сол кезде жазған, әкімші скриптерді міндеттерді орындау үшін шақыра алады, олай болмаған жағдайда жалықтыратын еді. Қарапайым мысал, сценарийді жазу, мәтіндік файл аттарын пайдаланушылардың мүмкіндігін және жаңа есептік жазба жасау үшін пайдаланушы атын қабылдау керек.

Жүйелік компьютерлерді, принтерлер және файлдық серверлерді енгізе отырып әкімшілікпен шектелуі мүмкін. Дегенмен, әкімшілендіру жүйесі әкімшілендіру желісіне таратуы мүмкін және тіпті веб-сервер, FTP-сервер, пошталық сервер және әкімшілік сервер деректер базасының қажеттіліктеріне байланысты және компанияның мөлшері мен мүмкіндіктері жүйелік әкімші (лер) де таратуы мүмкін. Ақыр соңында, жүйелік болады -*+

Пайдаланушыларды оқыту үшін әкімшілік қажет болуы мүмкін. Сондықтан, әкімшілендіру жүйесі үшін әр түрлі қажетті дағдылар болуы мүмкін. Жүйелік әкімші нақты жалпы міндеттерді қамтиды:

- Тіркелгілерді басқару: жаңа пайдаланушылардың ескірген тіркелгілерін пайдаланушылардың тіркелгендерін жою.
- Құпиясөз басқармасы: бұл барлық пайдаланушылар құпия сөздерді қауіпсіздік саясатымен келіседі (мысалы, құпия сөз ай сайын өзгертілуі тиіс, кем дегенде құпия сөздерді әліпбилік таңбасы жоқ болуы керек). Сіз таңқалуыңыз мүмкін, бірақ жүйелерде барабар құпиясөз басқармасы жоқ, көптеген пайдаланушылар құпия сөздерді қолданады (яғни, олардың құпия сөзді жай ғана батырмамен басу арқылы пернесін іске қосады). Басым көпшілігі бүгін сенімді құпия сөзді пайдалануды талап етеді: құпия сөздер, құрамында кем дегенде сегіз таңбадан, оның ішінде кем дегенде бір әліппесіз және / немесе бас комбинациясы және кіші әріптер және кем дегенде бірнеше айдың ішінде көп мәрте пайдаланылады, құпия сөздер бір рет өзгеріп отырады.
- файлды қорғау басқармасы: файлдар тиісті түрде қорғалған (мысалы, бұл сыртқы әлеммен маңызды құжаттар жазу үшін қол жетімді емес) файлдық жүйесі сақтық дер кезінде орындалатынына көз жеткізіңіз.
- Орнату және жаңа жабдықтардың ақауларын және жабдықтарын, соның ішінде желісін жою.

- Орнату және бағдарламалық қамтамасыз етуді қоса алғанда, жаңа операциялық жүйесін (ОЖ) жаңарту, түзету және бағдарламалық қамтамасыз ету ақаулықтарды жою.

- Пайдаланушылардың компьютерлер құжаттамаларын ұсыну, қолдау және оқыту.

- Жүйелік бағдарламалау деңгейде орындауына қарай қажет (әдетте көмегімен сценарийлер тілі емес, жазу үлкен көлемді қосымшалар немесе жүйелердің бағдарламалық қамтамасыз етуі).

- Қауіпсіздік: брандмауэрды қарастыра отырып монтаждау және техникалық қызмет көрсету, файлдар журналын көру үшін қандай да бір оғаш өрнектер эрекеті кезінде кіру, сондай-ақ зерттей отырып күдікті процестертер болуы тиіс емес, іске қосылмайды.

Көптеген жағдайларда желі әкімшілігі жүйелік әкімшіліктен бөлек болады. Бұл желілік әкімші барлық компьютерлік желілер аспектілеріне жауап береді. Желі әкімшісінің міндеттері физикалық белгілейтін кабельді қосу және желілік құрал-жабдықтармен қамтиды (мысалы, бағдарлаушы мен коммутаторлар). Желі әкімшісі, сондай-ақ жекелеген машиналарды теңшеуі тиіс, қарым-қатынас жүйесі арқылы араласу мүмкіндігі бар. Осылайша, жүйелік әкімші және желі әкімші кескіндеме файлдарды редакциялау, бағдарламалық қамтамасыз етуді орнату (желісімен байланысты) және осылайша, бұдан да әрі. Ақаулықтарды жою желісі және іздеу, сондай-ақ әкімшісінің желілерінің рөлі үшін, онда бұл жағдайда, іздеу және ақауларын жою мүмкін қиыстыру жеке және ақауларды жою (мысалы, кабель) және бағдарламалық қамтамасыз ету ақаулықтарын жою.

Сондай-ақ компьютерлік желіде қауіпсіздік аспектісі бар. Әкімшілік ретінде және желі әкімшілігі жүйелік брандмауэрлер жұмысын істей алады. Редакциялау конфигурациялық файлдарды, жазу скриптері мен бағдарламалық қамтамасыз етуді орнату және барлық бөлігінде желі әкімшілерінің міндеттері болады.

Басқа әкімшілік міндеттерді АТ-персонал қолдау көрсетеді. Қолдау әдетте екі нысанда шығарылады: оқыту және анықтамалық қызмет. АТ-инфрақұрылымында жаттығу. Бұл қарапайым заттарды қамту мүмкіндігі ретінде компьютерлік жүйені баптау, принтерлерді кіргізу, ортақ файлдарға қол жеткізу және қызметкерлерді оқыту мүмкіндігі, жұмысына байланысты бағдарламалық қамтамасыз етуді қалай пайдалануды үйрету. Адам құжаттаманы, пайдалы сайттарды (соның ішінде вики-беттер) және тіпті аудиовизуалды демкилерді құра алады, оқу сабақтары топта жүреді немесе жеке жұмыс жүргізіледі. Өйткені, оқыту тек қажеттілігіне қарай (жаңа бағдарламаны қамтамасыз ету,

жаңа қызметкерлер), ең көп қолдауға анықтамалық қызмет түрінде жеткізіледі. Істің мәні бойынша, бұл талап үшін, кім үшін қолжетімді болуға тиіс әрекет ету мәселелері, туындайтын кездейсоқтық болады. Көптеген ірі ұйымдар 24/7 анықтамалық қызметті ұсынады. Анықтама қызметінде адам ретінде жай тарату қалқанынан, АТ бағыттау үшін керек. Басқа жағдайларда, қолдау қызметінде адам осы мәселені тікелей, жиі телефон арқылы, бірақ кейде электрондық пошта арқылы немесе өзі шеше алады.

Ол қайда пайдаланылады? Бүгін АТ жаппай болып табылады. Айталық, әрбір адам ғаламшарда есептеу техникасының кейбір нысандары көмегімен ұялы телефондар мен планшеттер немесе үй компьютерлерін немесе оларды мектепте де, жұмыста да пайдаланады. Дегенмен де, көптеген АТ-бөлімінде АТ-қызметкерлерді жалдайды. Бұл ұйымдар бірнеше адамды немесе ондаған мың корпорацияны қамтуы мүмкін.

1.2-кестесі АТ – ның ірі пайдаланушылары мен АТ-ны қалай пайдалану керек екендігі туралы түсінікті қамтамасыз етеді.

1.2 кесте. АТ-ны ірі пайдаланушылар.

Кәсіпорын түрі	Типті қолдану
Бизнес	Электрондық коммерция, тұтынушылар; туралы жазбалар
Білім беру	Дәстүрлі есепке алу, қолдау және оқыту;
Ойын-сауық индустриясы	Сандық монтаждау, арнайы әсерлер, музыкалық композициялар, жарнама;
Үкімет	Іс-қағаздарын жүргізу, талдау жасау, ақыл-ой, ақпарат тарату;
Денсаулық/ауруханалар	Іс-қағаздары, медициналық құралдар, сақтандыру;
Құқық қорғаушы органдар	Іс-қағаз жүргізу, ақпаратты жинау және жіберу;
Өндіріс	Өңдеу, автоматтандыру, роботтар;
Зерттеу	Есептеу техникасы, ақпарат жинау

АТ-ны кім үйренеді?

АТ-мамандары ережеге сәйкес жағдайда жиі әзірленеді. Келесі сценарийді қарастырайық. Иллинойс штатының университетте Джо компьютерлік ғылымдар саласында бакалавр дәрежесін алды. Ол Чикаго фирмасына бағдарламалық қамтамасыз етуден бірден жұмысқа қабылданды, ол COBOL тілінде жұмысқа бағдарламашы ретінде жұмыс істеуге қабылданады. Дегенмен, ол 3 ай ішінде, жаңа келген жұмыскер бола отырып бастық одан сұрайды: " Сіздің Linux операциялық жүйе туралы білетініңізге сенімдімін, солай емес пе?"- дейді. Әрине, Джо Unix-ті бакалавр бөлігі ретінде дәреже алу үшін білген және былай жауап берді: "Әрине". Сондықтан босс былай деді: Джо "осы сәттен

бастап, мен сіздің аптасына 10 сағатыңызды насихаттауыңызды жұмсауыңызды қалаймын, Бұл Linux қолданатын жаңа компьютерлер желісі еді. Біздің файлдық серверлерге қосылуы мүмкін екеніне және оны қауіпсіз етуіне көз жеткізіңіз. "Джо басшылыққа оқуына аптасының 10 сағатын өткізді, Linux қорғаншылар мен жүйені бірге құрып орнату, операциялық жүйесімен ойнап және ақырғы қорытындыға келгенде, жұмыс істете алды. Кейін кейбір бастауыш вегетациялық ауырсыну жүйесінде пайдалану, көп қызметкерлер Linux платформасына көшті. Енді арада 9 ай өткеннен соң, ұйымның жартысы Linuxқа көшті, бірақ жүйе міндетті түрде өтуі тиіс емес. Әрдайым мәселе туындаған кезде, Джо әдетте, жауап беретін және түзетін адам ретінде болды. Босс оралып Джоға былай дейді: "Сіздің жүйеде істегеніңіз шебер жұмыс. Мен сізді жүйені қолдау үшін толық жұмыс күніне ауысуыңызды қалаймын. "Джо бұл үшін мектепке бармады, себебі онда кейбір дағдылар болды, сондай-ақ ол ақылды, еңбекқор жеке тұлға болатын (Ол университетті Computer Science Иллинойс! бағдарламасын аяқтау керек еді), ол бұл бастауда табысты болды. Оның орнына біреуді жалдап жүйесін қолдау үшін, Джоның осы ереже тұрақты негізінде жылжыту шешімі оңайырақ. Байғұс Джо, ол кодты жазғысы келді (дегенмен, CO-VOL мүмкін емес). Бірақ енді жалғыз код, ол Linux скриптер қабығын жазды! Дыбыс екіталай ма? Шын мәнінде, бұл өте кең таралған ертегі 1980-ші және 1990-шы жылдары, тіпті 2000-шы жылдары. Тек 2000-шы жылдардың ортасында, АТ-персоналдың рөліне сәйкес келуі үшін бұл АТ-оқу бағдарламасы компьютерлік ғалымдарды немесе адамдарды ғана компьютерлік аквариумистың орын алуына әзірледі.

Бірнеше, 2-жазғы техникалық жоғары оқу орындарының дәрежесі ұқсас "білікті" әріптестер болды, бірақ бұл колледждер ДК жөндеу және ақаулықтарды жою көбінесе бағытталған ұғымдар сияқты жүйелік және желілік әкімшілік емес болып қамтитын. Бүгін біз бұл адамдарды көруге, тек заманғы технология бойынша дайындық қана емес, сонымен қатар АТ-инфрақұрылымын қоса алғанда барлық көріністерін түсіну, теориялық сұрақтар, математикалық компьютерлер (бинарлық), компьютерлік жүйені құрайтын әртүрлі компоненттердің рөлі, бағдарламалау әдістері, операциялар деректер базасын, желілердің, интернет және арнайы білімдер, компьютерлік сот-медициналық сараптама сияқты. Жалпы АТ бағдарламасы жүйелік операциялық платформаларға кіріспені, тілдерді бағдарламалау және есептеу тұжырымдаманы қамтиды.

Біздің ойымызша, студент болу үшін, жұмыс тәжірибесі Windows және Linux немесе Unix болуы тиіс). Бағдарламалау тілдері - ИНГ тілдері сияқты Linux / Unix сценарийлер қабығын, қызыл немесе Ру-

thon және Java Script және құрастыру тілдерінде C, C ++ сияқты, Java немесе Visual Basic қамтиды. Тұжырымдамалар қамтиды, операциялық жүйелер мен желілерді өзіне қосады, бірақ оларды қосу шеңберінен веб-инфрақұрылымды, компьютер архитектурасын, бағдарламалық қосымшаларды (мысалы, бизнес-бағдарламалық қамтамасыз ету), сандық бұқаралық ақпарат құралдарын және деректерді сақтау, электрондық коммерцияны шығаруы мүмкін.

АТ-бағдарламалар түрлері

Дегенмен, 4-жастағы АТ деңгейі салыстырмалы жаңа болып табылады, ол сондай-ақ стандартталған. Әр түрлі университеттерге мұндай АТ-бағдарламаның көзқарасы әртүрлі дәрежеде ұсынылды. Мұнда біз жалпы тәсілдерді қарастырамыз.

Біріншіден, олар тармақталған компьютерлік ғылымдар градусы болып табылады. Жаратылыстану жұп дәрежесі бірге бар сияқты, өйткені көп қайталаулы, бұл екі пәнді қамтуы тиіс: аппараттық технологиялар, бағдарламалау, мәліметтер қорларын жобалау, компьютерлік этнографикалық желілер. Дегенмен, компьютерлік ғылым дәрежесі әрқашан қатты бағдарламалау айналасында және АТ дәрежесі аз қажет болуы мүмкін. Сонымен қатар, информатикада математика маңызды рөл атқарады, бірақ, мұндай математика саны АТ-да түсініксіз болып келеді. Бұдан әрі, ақпараты жүйелерінің басқару вариациясы бар. Ой-пікір мынада: ол оқытуды пайдалана отыруы тұрғысынан, көп қосымшалар, деректер базасында деректерді сақтау, ал технология негізінде бизнес-қосымшалар жатады. Электрондық коммерция, деректер базасы дизайны, деректерді интеллектуалды талдау, компьютерлік этика және мұнда заң жылжытылған. Сонымен қатар, АТ басқарумен байланысты курс тұжырымдамасын қамтуы мүмкін.

Сонда инжиниринг технология тәсілі бар, ол аппараттық пішінін төлем ақысы шоғырланады, дисктік жинақтағыштар, ӨКақаулардың құрылысын анықтау және жеке көріністері желілерін жою. Бағдарламалауға аз назар бөлінген, бірақ әлі де бағдарламалау компоненті бар.

Басқа тұрғысы – бұл компьютерлік желілердің негіздерін қамтамасыз ету. Бұл оқулық осы идеяға сүйене отырып, бірінші аппараттық бөлігі компьютерлік жүйесін, содан кейін операциялық жүйелерді ұсынады. Сондай-ақ, біз компьютерлік желілерді, бағдарламауды, компьютер және технологияны түсінуді сақтауымыз керек. АТ-түлегі АТ әкімшісі ретінде жұмыс істей білуі ғана тиіс емес, бірақ АТ-жүйесі дизайн сәулеті аппараттық құралдарын желіге жылғы бағдарламалық қамтамасыз етеді.

SIGITE, АҚМ бойынша арнайы топ мүдделеріне АТ-білім береді, АТ-моделі- оқу жоспары үшін пайдалы нұсқаулықтарды көрсетеді.

АТ- маман болу үшін, сіз міндетті түрде қатаң математикалық немесе инженерлік компьютерлік білім ғалымдары мен компьютерлік инженерлер, көп қабаттасатын талант иесі болуыңыз керек; ең маңызды талант дағдылары кінәраттарды іздеу болып табылады. Тағы бір сіздің жүйеңізде АТ-адам анықтайды. Бұл диагностикалық дағдыларды таза үйрету мүмкін емес. Сізде фондық білімдер және түйсік тәжірибесі болуы керек. Ең алдымен, сіз білуіңіз тиіс, ол қандай жүйе, қалай жұмыс істейді, Linux операциялық жүйесінің басқаруымен, компьютерлік желілер, веб-сервер немесе басқа. Тағы бір талантты иесі болу, бағдарламалық кодты жазу, барлық ықтималдығын жазу болар еді, шағын бағдарламаларға немесе сценарийлерге, инженер-бағдарламаушыға қарағанда, ірі масштабты жобаларға қатысады.

Сіз басқа адамдармен тіл табысуыңыз тиіс, сондықтан сіз мәселелерді түсінесіз, өз әріптестерімен тұтынушыларға хабарласқаныз үшін және өз кезегінде осы мәселелерді шешу жолдарын сипаттайды. Бұл телефон арқылы емес, жеке іс-қимыл болуы мүмкін. Сіз сондай-ақ техникалық жазу жағынан қабілетті болуыңыз керек. Сіз құжаттаманы жиі дайындау және есептемені ұсына аласыз. Және ақыр соңында, сіздің өз бетінше оқуға қабілетіңіз болу керек, өйткені технология үнемі өзгеріп отырады. Сіз білімді ертерек оқыту кезінде немесе өз бетінше білім алу да алсаңыз, ол бір жыл немесе екі жылдың ішінде ескіріп қалуы мүмкін. Сіз білімді мектепте немесе оқыту арқылы алуыңыз мүмкін, бірақ бір немесе екі жылда ескереді. Дегенмен, сіз үйренуді жалғастыру үшін іргетасты қалыптастырып алуыңыз керек. 1.3 кестені қараңыз, онда АТ-қызметкерлерден дағдыларды талап ету жоспарын бірінші орынға қояды.

1.3 кесте. АТ дағдылары.

Дағдылар	Сипаттары	Мысалдар
Ақаулықтарды табу және мәселелерді шешу	Мәселесін анықтап, оның себебін табу Шешім табу (ақаулықтарды жөндеу)	Дискілік кеңістік төмен өнімділік процессоры Вирустар және троян жұқпасы

Операциялық жүйені білу	Операциялық жүйе қосымшаларды жасау тіркелгі пайдаланушы қондырғылар жүйесін орнату	Linux нұсқасы Unix нұсқасы WindowsMacOS
Жүйе дәрежесіндегі бағдарлау	Жүйелік мониторинг Shell скриптілері үдерістерді автоматтандыру үшін конфигурациялық файлдарды басқару жүйелік қызметі	Bash, CSH скриптері DOS, рубин скрипты, C/ C++ бағдарламалары
Жүйе қауіпсіздігін қамтамасыз ету үшін тиісті қауіпсіздік жүйесі	Кейін немесе саясатты әзірлеу үшін мониторинг пайдаланушылар қауіпсіздік жүйесі	Орнату брандмауэр жүйесін орнату Вирусқа қарсы/ бағдарламалық қамтамасыз етуді қорғау зиянды файлдар бағдарламаларының сараптамасы, журналдар үшін басып кіруді дәлелдеу және жүйелі қауіпсіздігінің ілесуі
хардвер	Аппараттық құралдарды және жаңа жабдықтарды орнату Ақауларды жою, жөндеу немесе жарамсыз жабдықтарды ауыстыру	Дисктік жинақтауыш соңғы жаңартуларымен қауіпсіздендіру Процессорларды ауыстыру және дисктік жинақтаушыларды қосу, желілік кабельдер - желілік концентраторлар, коммутаторлар, бағдарлаушылар

Онда, бәлкім, АТ-студенті үшін прототиптік емес. Бірақ АТ-студенттің міндеті:

1. Компьютермен ойнау кезінде рахаттаныңыз, тек оны пайдаланыпқана қоймай, оның қалай жұмыс істейтінін жүйелі оқыңыз.
2. Өзіндік білу алуда разаттаныңыз- сіздің міндетіңіз- жаңа заттарды табу және олардың қалай жасалатынын білу керек.
3. Технология- бұл ғажайып, технологиядан қорықпаңыз, бірақ оны барлық формада қабылдаңыз.

4. Ізденіспен және ақаулықты жоюмен рахаттаныңыз.

АТ-студенті бағдарламалаудан рахат алуы маңызды емес. Шын мәнінде, көптеген студенттер АТ мансап ретінде таңдайды, өйткені олар алдымен компьютерлік ғылымнан бастайды, бірақ көп ұзамай пәндерді бағдарламалау ауыр екеніне көздері жетеді. Бұл АТ-студентке бағдарламау емес, бірақ бұл бағдарламалау кем қарқынды, ол негізінен шағын скриптерді жазуды талап етеді.

Мысал ретінде, студент информатиканы зерттеу барысында, мәтіндік жазу процессор жүйесі деректер базасын басқару немесе аудармашы тілі болса, АТ-студенті сценарийлерді автоматтандыру үшін жазып, есептік жазбасын жасап, пайдаланушы немесе тұтынушылар жағынан сценарийлерді жазып, кепілдік беру үшін, веб-нысаны толтырылған болса немесе серверлік сценарийлерді жазу, веб-форманы өңдеуді жаза алады. Айта кету керек, бұрынғыдай көп қырлы жүйелік әкімшілендіру позициясы қамтылмаған. Студенттер АТ-мансап ретінде ойламай, "ойнау" технологиясымен жұмыс деп ойлайды. Бұл қызықты, бірақ ол сондай-ақ АТ үшін мәселе болып табылады.

Қазірдің өзінде ақпараттық технологияларды хобби ретінде деп айтсақ та болады. Дегенмен, студент жұмыста жауапкершілікті ұстау үшін ондаған немесе жүздеген АТ-инфрақұрылымды ұйымдастыруға тартылады. Жабдық құны жүздеген мың доллар болуы мүмкін, бірақ бизнестің өзі миллиондаған доллар жасай алады. Осылайша, АТ маманы өз жұмысын шындап-қабылдауға дайын болуы керек- жүйелік қателер, тоқтап қалулар және т. б. ұйым үшін қымбат тұруы мүмкін. АТ маманы жүйелік әкімші міндеттері бар шеңберінен шығатын болуы тиіс. Осы үміттердің кейбірі төменде қарастырылды.

Жүйелік әкімші үшін облыс аумағынан хабардар болуы тиіс. Ең аз дегенде, жүйелік әкімші қауіпсіздік мәселелері қалай пайда болатыны туралы және олардан қалай қорғану туралы білуі шарт. Олар қамтамасыз етуде жүйеде вирустарды қамтуы мүмкін, желілікті басып кіруді, шабуылдардың бас тартудан қызмет көрсету, сондай-ақ SQL енгізе отырып шабуыл жасау. Сонымен қатар, жүйелік әкімші көшінен қалмауы тиіс, жаңа нұсқалары операциялық жүйе және / немесе серверді бағдарламалық қамтамасыз етуді, ол қолдайды. Дегенмен, жүйелік әкімші жаңа жабдықтармен, жаңа бағдарламалық қамтамасыз ету, сондай-ақ осы саладағы басқа да әзірлемелерге осыған ұқсас шеңберден алыстамауы керек. Жүйелік әкімшілік үшін, жаңа технологиялармен ілесе отырып, жаңа үрдістер, жаңа қауіпсіздік түзетулер, үздіксіз білім беру маңызды. Жүйелік әкімші өмір бойы оқушы мен өздігінен жүргізу болуы тиіс. Жүйе әкімшісі интернет форумдардан көрінуі мүмкін, бірақ үнемі әрі қарай жаңалықтар

технологиясын және өз зерттеулерарқылы бақылап-баптарында дайын. Сондай-ақ жүйе әкімшісі жаңа бағдарламалық қамтамасыз етеді және экспериментке дайын болуы тиіс, олар оны әлеуетті пайдалану үшін анықтайды. Жүйелік әкімші "шақыру бойынша" жұмыстан тыс уақытта жиі болады. Бір бөлеге ұшырағанда, жүйелік әкімші қол жетімді болуы тиіс. Жедел телефон шалу 3-те таңғы немесе демалыста әбден мүмкін. Дегенмен, әрбір қызметкер өз уақытында ережелерге лайық келісім-шарт әкімшісінің жүйесін, қайда баруға болатынын 24/7 қамтуы мүмкін. Мұндай сенімділік, деректерді ұйымдастыру файлдармен істемеуі мүмкін немесе орындау транс - әрекеттің бірнеше сағат бойы болмаған болуы мүмкін, миллион доллар шығын келтіруі мүмкін. Кейбір беделі компаниялар залал келтірілген шабуылдардан бас тарту жасалатын қызмет көрсету және қабілетсіздігі тез қалпына келеді. Жүйелік әкімші өзін әдепті ұстауы тиіс. Дегенмен, тап солай болса сіз қандай болар едіңіз, қандай да бір моральдық дилеммаға қақтығысасыз ба? Мысалы, сіз алаңдаушылықпен бұл ұйымның тым көп қызметкерлері электрондық пошта үшін жеке заттарын пайдаланады, сондықтан электрондық пошта арқылы басшы іздеу салады. Мұны қалай сезінуге болады? Енді ойлап көріңізші, бұл саясат ұйымы қызметкерлерді пайдалана алады деп мәлімдейді, электрондық пошта бойынша осы уақытқа дейін жеке мақсаттар үшін, электрондық пошта құпия сөзсіз аша алмайды. Бұл жағдайда, іздеу арқылы жұмыскерді сізден сұрайды.

Ай 1, бұл өзгертуді сіз қалай сезінесіз? Алдын ала рахмет!

Әдепсіз тәртіп келесідей болуы мүмкін:

1. Басқаларға тыңшылық жасау (электрондық пошта, веб-браузер, файлдарды қарау әдеттері).
2. Компьютерлік жүйелерге кіруге бэкдор шотын орнату.
3. Бағдарламалар немесе файлдарды заңсыз жүктеу немесе көтермелеу/осыны істеуге басқаларға рұқсаттама беру.
4. Жүйенің әкімшілік рұқсатына байланысты ұрлық немесе қасақаналық жасау

АТ-инфрақұрылымы

Ол компьютер айналасында шоғырланады. Сіз бүгін компьютерді қолдандыңыз ба? Сіздің жұмыс үстеліндегі (немесе ноутбук) компьютерде электрондық пошта тексеру үшін отырдыңыз, яғни ол сіздің компьютерді пайдаланғаныңыз? Сіздің ұялы телефоныңыз – бұл да сіздің электрондық кітапты оқу құралы сияқты компьютер. Олар үстелдік құрылғыларға қарағанда, бірақ олар компьютерлер болып

табылады. Компьютерлік компоненттер сіздің автокөлігіңізде және сіз жүретін қала көшелерінде бар. Сіз жұмыс істейтін немесе оқитын ғимаратта компьютерлерді жұмыс істеуге немесе оқуға пайдалануы мүмкін, сондай-ақ жарықтандыру мен ауа баптағышпен басқару үшін қолданады. Иә, егер біз мойындамасақ та, компьютерлер айналамыздағы барлық болып табылады. Біз компьютер бөлігі электрондық жабдықты болуын анықтаймыз, ол RUN - бағдарламалардың өзара әрекеттесу пайдаланушымен (енгізу-шығару) деректерді сақтауға қабілетті. Бұл міндеттер ЕОШС (енгізу, өңдеу, шығару, сақтау) циклдеп жиі атайды. Жалпыәмбебап компьютері болып табылады, ол кез-келген бағдарламаны іске қосады. Компьютерлер кез келген бағдарламаны іске қосатын, жалпы мақсаттағы құрылғылар болып табылады. Мысалы, сіздің iPod музыканы ойнатуға қабілетті; онда пайдаланушы интерфейсі бар, және күнтізбе үшін, тәулік уақытын көрсету үшін және де бірнеше ойындар қатарын көрсету үшін аздаған қосымшалар болуы мүмкін. Сіздің ұялы телефоныңыз да көп қосымшалар саны бар, бірақ ол бағдарламалардың көпшілігін жұмыс істете алмайды. Компьютер жалпы мақсаттағы едәуір дәрежеде негізделген, оның сыйымдылығының және бағдарламаның іске қосылуы үшін арнайы процессор жасалған. Компьютерлер мөлшері бойынша ерекшеленеді және мүмкіндіктері: супер компьютерлер болуы мүмкін, бөлме үшін үстелдік құрылғыларды толтыру, өте ауыр, бірақ арналған портативті емес, портативті компьютерлер, блоктарды, олар жарық сияқты, мүмкін, ауыр оқулық, портативті құрылғылар сияқты ұялы телефондар, mp 3-плеерлер. Мобильдік құрылғы мен үстел немес портативті құрылғылардың арасындағы айырмашылық - оның перифериялық құрылғылардың түрлері қол жетімділігінде жатыр (сенсорлы экранмен салыстырғанда толық көлемі пернетақта және тышқан, 2-дюймді экранмен салыстырғанда 20 дюймді монитор), оперативті жад көлемі және қаты дискті сақтау кеңістігі, сондай-ақ сіздің есте сақтауға қабілетті құралыңыз, мысалы, USB порттары арқылы немесе оптикалық дисктің дисководы арқылы өтетін флэш-жинақтаушылар.

Компьютерлер

Біз компьютерлердің неден тұратынын толығырақ келесі тарауда қарастырамыз. Қазіргі уақытта, біз компьютерлерді неғұрлым жалпы түрде қарастырамыз. Компьютерде электрондық, бағдарламалық құрылғылар бар. Бағдарламасын іске қосу үшін, құрылғы процессоры қажет [орталық процессор (CPU)], есте сақтауға арналған бағдарламалар

мен деректер, мүмкіндіктер енгізу және шығару, сондай-ақ мүмкін ұзақ мерзімді сақтауға және желілік мүмкіндіктерді (соңғы екі міндетті болып табылмайды) анықтау, компьютерлер ғана емес үстелге қойылатын және портативті құрылғыларды қамтиды, серверлер, әмбебап компьютерлер және суперкомпьютерлер, нетбуктер, ұялы телефондар, компьютер ойын приставкалары, mp3-плеерлер және кітаптар (мысалы, жағады). "соңғы екі жағдайларда арнайы құрылғы және әмбебап қолданылатын, олар тек бірнеше таңдаулы бағдарламалар болып табылады. Тарихи компьютерлер ұғымы жойылды. Бүгінгі таңда біз компьютерлермен бүкіл әлемде біргеміз.

1.1-сурет кейбір компьютерлер ауқымын суреттейді. Үлкен мониторы бар үстел блоктары және жүйелік блоктар, заманауи үлкен экраны бар портативті компьютерлері сияқты жалпы болып табылады. Портативті құрылғылар одан да танымал болып табылады, оларда дербес қалта компьютерлері (ДҚК), ұялы телефондар мен электрондық кітаптар бар. Мониторлар - бұл жазық экрандар болып табылады.

Біз енді қазіргі таңда үстелдік компьютерлерімізге дағарадай мониторларды қажет етпейміз. Дегенмен, жүйелік блоктың көптеген құрылғылары, дискілі жинақтаушылары және басқа да ауыр компоненттері бар, сондықтар ол дағарадай болып табылады. Біз портативті компьютерлерді қолданған кезде, кейбір перифериялық құрылғыларды қиямыз. Сандардың ауыспалы құрылғылар қолжетімділігіне көшейік. ДҚК жағдайында лэптоп, ноутбуктер,



1.1-сурет Компьютерлер типі

чиптер және жүйелік тақта, сондай-ақ басқа да нысандарын сақтау, олар үлкен емес орындарда орналасуы тиіс. ДҚК үшін, бұл жай ғана сымсыз карта, ұялы телефон желісіне қосылуға рұқсат береді (Wi-Fi мүмкін). Портативті компьютерлер мен ноутбуктер үшін бәлкім қатты диск болуы мүмкін. Лэптоп те, сондай-ақ бәлкім, оптикалық дискілерде жинақтауыш болуы мүмкін.

Компьютердің негізгі компоненті процессор болып табылады. Процессор рөлі біз іске қосқан компьютерде бағдарламалық қызметті орындайды. Бағдарламасын іске қосу үшін осы компьютерде бағдарлама жасалуы тиіс. Жинақ процесі аударма тілі болып табылады, ол бағдарламасын оқыған түрінде түрлендіреді (айталық, Python немесе Java) компьютердің машина тілі. Компьютер осы машина тілінде жазылған бағдарлама негізінде ғана жұмыс істей алады. Біз бұл тұжырымдамасын толығырақ, сондай-ақ процессордың рөлін 2-ші тарауда қарастырамыз. Кейінірек біз осы кітапта бағдарламалау тілдерін қарастырамыз. Процессордан бөлек, компьютерлер қоймаға мұқтаж болады. Екі түрі бар (Storage - қысқа мерзімге сақтау және ұзақ уақыт сақтау. Қысқа уақыт сақтауға жиі құрылғысы- бұл жедел есте сақтау құралы (ЖЕСҚ). Өкінішке орай, жедел жады бірнеше әр түрлі типті сипаттауы мүмкін. Біздің қазіргі заманғы компьютерлерде үш жад нысаны бар: динамикалық ЖЕСҚ (яғни, әдетте біз оны негізгі жадты деп атаймыз), статикалық жедел жады (кэш-жад пен тіркелімдер) және ҚЕСҚ (қалыпты есте сақтау құрылғысы). Біз 2-тарауда осы жад түрлерінің арасындағы айырмашылықты қарастырамыз.

Қазіргі уақытта, дәп қазір «есте сақтау үшін» барлық үшеуін қарастырамыз. Жад (динамикалық ЖЕСҚ) микросұлбалардан тұрады. Динамикалық ЖЕСҚ жылдам кіруге мүмкіндік береді. Дегенмен, кейбір портативті құрылғыларда үлкен жедел жад жоқ. Оның орнына, олар сақтау үшін флэш-жадты қолданады, онда көлем бойынша неғұрлым шектеулі болып табылады. Ұзақ сақтау үшін қатты дискілер жиі қолданады, бірақ сондай-ақ оптикалық диск, флэш-жад, магнитті таспаны қамтуы мүмкін. Әлдеқайда көп қуатты сақтауға қарағанда ұзақ, қысқа мерзімге сақтау, сондай-ақ, әрқашан қосымша орын сақтауға болады, біз ұзақ мерзімді сыйымдылығын шексіз қарастыру. Десктоп пен лэптопты типтік қысқа мерзімге сақтау 1-8 Гб диапазонында болады. 1 Гбайт 1гигабайтты білдіреді, яғни ол 1Гб - 1 миллиард рәміздерді сақтай алады. Типтік кітап, бәлкім, шамамен 250 мыңнан 1 миллион дейін рәміздерді құрайды. 1 Гб кемінде 1000-нан кем емес кітапты (сурет) сақтайды. Қатты дискілер енді 1 ТБ, немесе 1 терабайт, 1 триллион байт құрайды. Әлбетте, ұзақ мерзімде сақтау қысқа мерзімдік

сыйымдылықты сақтауға қарағанда әлдеқайда көп. Кейбір жалпы мөлшерін сақтау 1.4. кестеде келтірілген. Біз сақтау мөлшерлерін 2 және 3 тарауларда қарастырамыз.

1.4 Кесте. Сақтау мөлшерлері.

Мөлшер	Мәні	Мысал
1 бит	0 немесе 1	Ең төменгі бірлік сақтау, 1 қара және ақ пиксель немесе 1 ақиқат / жалған мәнін біріктіруге көп бит сақтай алады жасау үшін не маңызды
1байт (1B)	8 бит	Біз 0-ден 255 немесе -128 ден 127 санын сақтауға немесе бір белгі (әріптер, сандар, тыныс белгісі)
1 word	32 немесе 64 bits	Деректер санының немесе команданың бір бөлігі сияқты бағдарламасы
1 кб	1024 байт	Жад көлемінде блокты сақтау
1 мб	~1 млн байт	Шағын сурет немесе үлкен мәтіндік файл, mp3 файл әндер кетуі мүмкін, 3-тен 10 МБ, 50 минуттық 350 МБ телешоу жоғары сығылған қажет болуы мүмкін
1 гб	~1 блн байт	әндер мен суреттер кітапханасы, ондаған кітаптар, бірнеше DVD гигабайт дискілік кеңістікті (4-8 ГБАЙТ) талап етеді
1 тб	~1 тлн байт	Фильмдер кітапханасы

Сақтаудың бізге екі формасы қажет пе ? Бұл құрылғы түріне байланысты болады және сіздің болжамды пайдалануды, бірақ жалпы біз екеуіне мұқтажбыз. Не үшін? Бағдарламасын іске қосу үшін, бізге бұл бағдарламаны кеңістік сақтау үшін жүктеу керек, жылдам түсіндіріледі. Әлдеқайда RAM қарағанда ұзақ сақтауға баяу, сондықтан, сіз өте баяу компьютерді қысқа мерзімге сақтауға дайын болуыңыз қажет. Екінші жағынан, қысқа мерзімге сақтау әлдеқайда шектелген қуатымен үлкен бағдарламаға біз жүгіреміз. Біз сондай-ақ өте жиі үлкен деректер файлдары (музыкалық файлдар, фильмдер және т.б.), осылайша, біз тек қана қысқа мерзімге сақтауға сүйенеміз. Портативті құрылғылары ұсынады, олар жиі пайдаланады, флэш-жады орнына жедел жады әкеледі, бәсеңдеуіне уақыт қол жеткізуді салыстырғанда үстелдік/портативті компьютерлер және олар шектелген кеңістік ұзақ мерзімде сақтау (егер ондайлар бар болса) талап етеді, қозғалыс пайдаланушының файлдар арасындағы қалталық құрылғылармен және тұрақты үшін сақталатын. Басқа жылдамдығы және сақтауға арналған сыйымдылықтар арасындағы айырмашылық жады бар және ұзақ уақыт сақтауға, басқа олардың өзгергіштігі дифференциялық фактор болып табылады. Ұшатын термин, сипаттау кезінде жад көрсетеді, қореккөзі ажыратылған болса, жадты сақтауға болады. Жедел жады (DRAM) және кэш/тіркелімі жад (SRAM) ұшпа нысандарда жад болып табылады. Содан кейін сіз қоректенуді өшіріңіз, мазмұн жоғалады. Міне, сондықтан, сіз компьютер жадын қосасыз, бастапқыда бос және "жүктеу" құрылғысы процесі арқылы өтуі тиіс. Энергияға тәуелсіз жад ROM дискісін камтиды, флэш-дискілер, қатты диск және оптикалық диск. Энергияға тәуелсіз жады шексіз ұзақ мазмұнды сақтайды. ПЗУ жағдайда мазмұн ешқашан жоғалмайды. Флэш-жинақтаушы және дискілер жағдайда мазмұн сақталады, олардың жоюын сіз шешіңіз. Компьютерлер перифериялық құрылғылар талап етеді (дегенмен дұрыс сөз, бәлкім бұл пайдаланушыға ыңғайлы болу үшін, біз айта кетсек перифериялық құрылғылар деп аталады). Перифериялық құралдар "шетінде", бірақ бұл жағдайда компьютердің, әдетте, сілтеме перифериялық құрылғылар, олар компьютерден тыс немесе нақтырақ, жүйелік блок. Жүйелік блок қорабы құрамында ана ақылы (онда орналасқан орталық процессор және жады) және диск жетек бірлік болып табылады. Перифериялық құрылғылар тым үлкен болып табылады, жүйелік блоктың ішінде сия немесе құрылғылар, олар адам пайдаланушылармен қол жетімді болуы тиіс (Сурет 1.2). Бұл құрылғылар біздің құрылғыларды енгізу және пернетақта-шығару, тінтуір, нүкте жолдар, трекбол немесе джойстик, монитор, принтер,

дыбыстық колонкалар, қалам және планшет (жазба немесе жарықтық қауырсын және т.б. осы құрылғылардың енгізу-шығару (ретінде белгілі, I/O құрылғылар ұжымдық), адамдар емес болады, өзара іс-қимыл мүмкіндігін белгіледі. Егер барлық кіріс мәліметтер файлдың дискідегі барлық шығу мәліметтері ұқсас болады түседі, сақталатын файл дискіде болуы мүмкін емес, компьютер үшін өзара ешқандай қажеттілігі іс-қимыл адам. Бірақ адам, сайып келгенде, бұл бағдарламаның не істегенін білгісі келеді.

Перифериялық құрылғылар арасында байланыс құрылғылары бар. Байланыс құрылымы – бұл компьютердің басқа компьютерлермен деректермен алмасуға мүмкіндік береді. Бұл құрылғылар әдетте модемдер болуы мүмкін, олар талап телефон желісі қосылғанын талап етеді (немесе, кабельді теледидар, коаксиалды желісі болуы мүмкін) немесе сымсыз болуы мүмкін. Барлық портативті компьютерлер бүгінгі таңда сымсыз модемдермен келеді, сонымен қатар, үстел компьютерлері бірліктері сымды немесе сымсыз модем болып келеді. Алайда, компьютерлер желіге қосылған жағдайда (LAN), компьютер желіге қосылуды немесе модемге қосылуды сұрайды. LAN қосу желілік карта көмегімен жүзеге асырылады, әдетте Ethernet-плата. Егер де



1.2-сурет Компьютердің перифериялық құрылғылары

жоғары жылдамдықты желілер туралы айтатын болсақ, онда желілілік карта неғұрлым жоғары қабілетке ие (деректерді жіберу жылдамдығы), яғни модемге қарағанда.

Біз енді әрі қарай оқулықта компьютерлік желілерді қарастырғанда, сымды және сымсыз модемдерді және желілік карталарды оқып үйренеміз.

Біздің компьютердің мәні жөніндегі білімдерді қысқаша мазмұндайық. Компьютер мәнінің жиынтығы әртүрлі құрылғыларды қамтиды, әрқайсысының міндеттері бар. Типтік компьютер мыналарды қамтиды:

1. Жүйелік блок, онда:

а. жүйелік тақта, ол қамтамасыз етеді:

1. Процессор

2. CPU салқындату блогы

3. Қосымша процессорлар (мысалы, сызба жұмысы үшін) қажет

4. ОЗУ, ПЗУ үшін жадтың микросұлбалары

5. перифериялық құрылғыларды қосу үшін ажыратқыштар (кейде порттар деп аталады)

6. Перифериялық құрылғылар карталары кеңейту үшін слоттар

7. ROM BIOS жүктеу үшін және базалық жүйесін енгізу және шығару нұсқаулығы.

8. Электрлі қуат беру жалғағышы

б. Жинақтауыш

в. Вентиляциялық құрылым

г. Қоректену

2. Монитор және пернатақта

3. Әдетте, кейбір нысанды көрсеткіш құрылғысы (тышқан, трекнүктесі, трекбол)

4. Бағаналар (қалауы бойынша)

5. MODEM немесе желілік карта (олар әдетте жүйелік блок ішінде орналастырылады, біріне қосылған слоттарды кеңейту)

6. Принтер (қалауы бойынша)

7. Сыртқы қатты диск және таспа жинақтауышы сияқты сыртқы деректерді сақтау құрылғылары.

2-ші тарауда фото орналасқан, ол жоғарыда аталған көптеген компоненттер суреттейді. Енді біз «компьютер» деп аталатын түсінікті анықтадық. Бірақ компьютер-бұл тарихтың бір бөлігі ғана. Бағдарламалық қамтамасыз ету, жалпы компьютерсіз мүмкін емес. Адамдардың көмегінсіз компьютер ешқандай бағдарламаны іске қоса алмайды. Осылайша, біздің компьютерлік жүйе осы компоненттерді қамтиды.

Бағдарламалық қамтамасыз ету

Бағдарламасын іске қосу үшін компьютер мағынасы қандай? Бағдарламаларсыз компьютерде ештеңе жоқ. Бағдарлама белгілі және сондай-ақ бағдарламалық қамтамасыз ету (ажырата білу үшін оның жеке компоненттерін, компьютердің аппараттық қамтамасыз ету), білдіреді нұсқаулықтарының тізімі, егжей-тегжейлі компьютерге не істеу керек. Бұл нұсқаулықта Java немесе Python тілінде сияқты бағдарламалау жазылуы. Бағдарламалау тілі нұсқаулықтың болуы тиіс сипаттамасы. Мысалы, егер сіз компьютерге енгізу үшін екі санының пайдаланушыдан және шығарудан біреуі көбірек, сіз жай ғана айта аласыз "Input екі нөмірлері мен екі көлемін шығару." Оның орнына әрекеттерін сіз сипаттайсыз, алгоритмі орын ретінде болуы тиіс, сыну нұсқаулықта арналған қадам. Нұсқаулықта бағдарламалау тілінде жазылуы тиіс. Мысалы, осы тармақта сипатталған мәселе төрт кезеңнен тұрады: Kіру number1 Input number 2 осы екі сандарды салыстырсақ, біріншіге карағанда, екіншісі number1 олай болмаған жағдайда шығу number 2 болады.

C тілінде, бұл келесі түрде көрсетіледі:

```
scanf("%d", &number1);
scanf("%d", &number2);
if(number1 > number2) printf("%d is greater", number1);
else printf("%d is greater", number2);
```

PRINTF командасына енгізеді, олар маңызды мәнді немесе хабарламаны шығарады. Егер команда үшін екі мәнді салыстыру және шешім қабылдауды пайдаланылады. Кейбір синтаксис C тән данасын "number1" және "number 2" Scanf операторлар, пайдалану нүктелі, үтір, аяқтау үшін нұсқаулар, сондай-ақ % D пайдалану. Әрбір өз синтаксисті болады және көптеген жағдайларда, синтаксис болуы мүмкін, оғаш біреу үшін емес, кім бағдарламашы болып табылады немесе жоқ екендігін білдім. Бағдарламаның тізімін операторлар ғана орындайды. Сондай-ақ бағдарлама, әртүрлі талапты анықтау. Олар хабарландырулар, айнымалылар, функцияларды немесе әдістерді және сыныптарды қамтуы мүмкін. BC біз number1 анықтаймыз және number 2 ретінде айнымалылар кодында жоғары пайдаланылады. Бұл мысалда, олар бүтін сандар ретінде жарияланады. Көптеген нысандары бағдарламалық қамтамасыз ету, бірақ біз тұтастай алғанда, оларды болады екі санаттағы бөлуге: жүйелік бағдарламалық қамтамасыз ету (операциялық жүйе

және қолданбалы бағдарламалық қамтамасыз ету (бағдарламалар, олар орындау үшін біздің міндеттері сияқты мәтіндік процессорды бастаймыз, интернет-браузер немесе компьютерлік ойындар). Әдетте, біздің бағдарламалық қамтамасыз ету инженер-маман бағдарламалық қамтамасыз ету жазылған. Алайда, сізде бағдарламалауға үйреніңіз, егер сіз қаласаңыз сіз өзінің жеке бағдарламалық қамтамасыз етуді жазыңыз. АТ-студент ретінде қысқа кесектері жазу кодын, скриптертерді үйреніңіз. Сценарий үшін қолдау немесе операциялық жүйесінде немесе қосымша пайдаланылуы мүмкін. Мысалы, сіз сценарийін Bash қабығын жазуға қызметі үшін автоматты түрде сияқты пайдаланушы тіркелгілерін жасау үшін жаңа топ пайдаланушыларды қолдау маңызды. Сіз жазу серверлік скрипт немесе Perl тексеру үшін URL мекен-жайын қауіпсіздігіне төнетін қауіп-қатерлерді веб-серверде болуыңыз керек.

Қолданушылар

Бұл компьютердегі пайдаланушы процестер. "Мұны қазір, бұл кейінірек." Бізге қажет, өзара әрекеттесетін бағдарламалар жұмыс істейді. Бұл интерактивтілік көмегімен енгізу/шығару құрылғыларды жүзеге асырылады. Бүгін біз соншалықты интерактивтілік, бәлкім елестету, пайдалана отырып, компьютерлерді үйрену мүмкін емес. Бірақ бұрынғы заманда (1940-1970-ші жылдары), көпшілігі, егер барлық адамның өңдеу мүлдем араласуынсыз жасалды. Қолданушы көрсеткен бағдарламасының көзі, кіріс деректер, көрсетілген орналасқан жері, шығу бағдарламасын қайтаруға жіберді. Қолданушы нәтижелерін көресіз, қалай ғана компьютер бағдарламасын іске қосты, ол дереу немесе бірнеше сағат өткеннен соң болар еді. Пайдаланушылар уақыт өте келе ұлғайды, сол технология жетілдірілді.

Ең ертедегі инженерлер пайдаланушылар компьютерге бағдарламауды салған. Компьютерлер соншалықты күрделі және қолжетімді болады. Компьютерлік шығыстар ретінде рұқсат етіледі, ұйымды сатып алу (миллиондаған доллар), компьютерлер пайдаланушылар қызметкерлері болды, сонымен қатар арнайы даярлықпен оларды пайдалану. 1970-ші жылдары алғаш рет шығарылған дербес компьютерлер келуімен жағдай өзгере бастады. Бірақ бұл емес, осы уақытқа дейін операциялық жүйесі windows бірге болған жоқ, бұл пайдаланушылар компьютерлерді қолдануды үйренуге алар еді. Ақпараттық технологиялар іс жүзінде жоқ. Міне, бүгін ол жалпы болып табылады, кез келген компьютерді әрбір адам пайдалана алады. Шын мәнінде, компьютерлер, әдеттегі құбылыс, бұл адамдар түсіну мүмкін емес, олар компьютер эстафетасын немесе

қосымшаны іске қосып, өз смартфонда GPS қойылады.

Бүгінгі таңдағы біздің көзқарасымыз

Біздің ойымызша бүгінгі компьютерлер қолдануға оңай болады. Олар қымбат құрылғылар, жеңіл, бөлмеге көбірек таза үй-жайларда талап етіледі, арнайы қоршаған кондиционерлер қолданылады. Адамдар іс жүзінде компьютерге жанасуға; олар өзара компьютер терминалдары іс-қимыл арқылы желі. Дербес компьютерлердің көмегімен, есептеу машиналарының жеке тұлғалар үшін қол жетімді және көптеген адамдар компьютерлер өздерінің меншікті үйлерінде болуға байланысты. Телебайланыстық, жергілікті желісі арқылы немесе телефон желісі, адамдарға біріктіруге өз компьютерлер мүмкіндік берді, бір-бірімен алмасу, электрондық пошта, хабарлар, файлдар, бағдарламалар және т.б. 1969 жылы интернет алғаш рет енгізіліп, коммерциялық пайдаланушылар үшін 1990-шы жылдардың ортасында үй компьютерлер қол жетімді болды. Осы кезең басында, адамдар интернетке қосылу арқылы баяу модем жөніндегі және телефонға қол жеткізеді. Бірақ соңғы 15 жыл ішінде, телебайланыс толығымен өзгерді. Енді бізде сымсыз қатынау, жоғары жылдамдықтағы интернет-қосылыстар, ұялы телефондар және тағы басқаларды пайдаланамыз. Бүгін компьютерлерді оңай сәйкестендіруге болады. Олар көп емес, шектеулі мейнфреймдер немесе үстелдік құрылғылар. Сіз желілік компьютер немесе ноутбук, планшетті компьютер, қалта компьютерін пайдаланасыз. Бізде тіпті аз мөлшерге қарағанда бар құрылғылар, қол құрылғыларды пайдаланатын процессорлары мен жады бар. Біздің қосу бірдей өзгерді. Сіздің байланыс құралдарына қол жетімділік осымен шектелмейді, телефон порты сіздің үйіңізде жұмыс атқарады. Сымсыз байланыс көмегімен, сіз кез келген жерде, сіздің үйіңізде немесе шалғайда қол жеткізуге болады. Сіз кофе-брейкті қалайсыз ба? Мәселе жоқ, Starbucks кафесіне барамын және егер сіз ноутбук арқылы интернетке кіре алсам. Немесе дискті қабылдай отырып? Егер сіз әлі интернетке ұялы телефон арқылы (сіз ақылға қонымды жақын мұнарадағы ұялы телефонға кіргенше). Біз әлем біріктірілуі арқылы дерлік бір сәттік байланысқа қарамастан, біз мұнда тұрамыз. Біз әлемде миллиардтаған процессорлар жұмыс атқарады. Біз пайдаланған миллиондаған компьютерлер санау үшін, бірақ бүгін ондаған миллиард процессорлар бар, көпшілігі оның ішінде бір-бірімен өзара іс-қимыл жасайды. Бұл кіріспе АТ ретінде қызмет атқарады, бұл мәтін нүктесі болып саналады. Басшыларының алған бойы көптеген байланысты АТ тұжырымдамаларын біз зерттедік. Біріншіден, біз

компьютерлік компоненттерді алуға, түсінуге, олар қалай жұмыс істейтініне және біз оларды компьютерге қалай қосуға болатынына назар аударайық.

Сондай-ақ, байланысты екілік атты санының және біз екілік файлды қалай пайдаланамыз. Бұл мынадай екі тарауы компьютерлік және ұйымдастыру екеулік жиі материалмен түсіп кетуі" бойынша информатикадан оқу бағдарламасын оқимыз. Олар АТ-адамға қарағанда процессор және ана төлемақысына келгенде, компьютерлік техниканы жақсы түсінеміз. Берік іргетасы бола тұра, компьютерлік компоненттерді жасау және олардың қалай қабілетті жұмыс істейтінін, оның қажеттілігін түсінуіңіз тиіс, қашан жедел жады немесе процессорды қалай бағалауды арттыру. Екілік файлды қосу бұл мәтінде негізінен тұжырымдамаларын қолдау, компьютерлік желілерде табылған. Фокус ауысуда жүйелік бағдарламалық, яғни операциялық жүйелерді қамтамасыз ету. Екі файлдық ең көп таралған платформалар мен операциялық жүйелерді қарастырайық: Windows (Windows 7) және Unix (Red Hat. Scanf командасын енгізеді, мән маңызы немесе хабарлама PRINTF командасы шығарады. Команда үшін екі мәнді салыстыруды пайдалану және шешім қабылдау. Кейбір синтаксис C тән данасын және "number1" және "number2" Scanf операторлар нүктелі үтір аяқтау үшін нұсқауларды, сондай-ақ % пайдалану. Әрбір тіл синтаксис болады және көптеген жағдайларда синтаксис өте оғаш біреу үшін емес болуы мүмкін, кім бағдарламашы болып табылады немесе жоқ екендігін білдім. Бағдарламаның тізімін ғана орындайтын операторлар. Бағдарлама сондай-ақ, әр түрлі анықтауды талап етеді. Олар хабарландырулар айнымалыларды, функцияларды немесе әдістерді және сыныптарды қамтуы мүмкін. Мысалы біз анықтау бойынша C number1 және number2 ретінде кодында жоғары айнымалылар пайдаланылатын. Бұл мысалда, олар бүтін болады,сандар ретінде жарияланады. Көптеген нысандары бағдарламалық қамтамасыз етеді, бірақ біз тұтастай алғанда, оларды екі санатқа бөлуге болады: жүйелік бағдарламалық қамтамасыз ету (операциялық жүйе және қолданбалы бағдарламалық қамтамасыз ету (бағдарламалар, оларды орындау үшін біздің міндеттеріміз сияқты мәтіндік процессорды, интернет-браузер немесе компьютерлік ойындардыбастаймыз).Әдетте, біздің бағдарламалық қамтамасыз ету жазылған инженер-маман. Алайда, сіз бағдарламалауды үйреніңіз, егер сіз қаласаңыз өзінің жазу жеке қамтамасыз ету бағдарламалықты қолдана аласыз. АТ-студент сіз қысқа кесектердің кодын жазуға, скриптерді үйреніңіз. Сценарий пайдаланылуы үшін немесе қосымша операциялық жүйесінде қолдау мүмкін. Мысалы, сіз сценарийдің қабығын жазуға

Bash қызметін қолдау үшін автоматты түрде сияқты тіркелгілерін жасайтын жаңа топ пайдаланушылар. Немесе сіз жазу серверлік скрипті Perl-де тексеру үшін URL қауіпсіздігіне төнетін қауіп-қатерлерді веб-сервер мекенжайында болуы. Біз, олардың қалай көрінетінін, қалай біз оларды пайдаланамыз және параметрлерін қарсы қойып салыстырамыз? Операциялық жүйесімен бірге файлдық жүйені пайдаланушылардың есептік жазбаларын қамтиды және ағыммен миссиялардың, процестер мен процестерді басқару қызметтері. Біз сондай-ақ, екі linux - нақты тақырыптарды қарастырайық: Bash қабығының және Linux тұрақты өрнектерді пайдалану.

Бірнеше мәтінінде әр түрлі әңгімелер қаралады. Компьютерлік аппараттық құралдар эволюциясы, операциялық жүйелердің эволюциясы, компьютерлік бағдарламалау, Linux тарих ретінде және Windows интернеттің барлық жабылғанын көреміз. Дегенмен, АТ-адам мүмкіндігі үшін қажетті емес болып табылады. Орнату фонын көріп, технологиясын өзгертуге көмектеседі, сондықтан, АТ тез өзгерістер туралы түсіндіреді. Сонымен қатар, өткенді түсіне отырып, бұл сізге көмектесе алады. Соңғы АТ басқа тақырыптары басшыларының жинағын қамтиды. Компьютерлік желілер бағалануы бірнеше әр түрлі көзқарастар болатын. Логикалық құрылымы, желінің физикалық табиғаты желілер, желілік хаттамалар, жазу ретінде берілген деректер өңделуі тиіс және кейбір ең көп таралған желілік бағдарламалық қамтамасыз ету барлық тарауда қаралады. Бағдарламалық қамтамасыз ету басқару түрлері етуді сипаттайды, қолжетімді және толығымен ақпаратты қамтиды, компьютерлік жүйені орнату бағдарламалық қамтамасыз ету, атап айтқанда, Linux бастапқы коды ашық. Тағы бір тарау бағдарламаланы ұсына отырып, жазу скриптер ретінде қабығына Linux және DOS мысалдар арналды. Соңғы тарау ақпараттық жақты мәтінді қамтиды. Бұл тарауда біз ақпаратпен қамтамасыз ету басқаруды, ақпараттық қауіпсіздік ұғымдарды сияқты қарастырамыз. Қорытынды мемлекет мәтінін қарастыра отырып, АТ-мансап және әр түрлі тақырыптарға байланысты АТ-мамандар бұрады.

Әрі қарай оқу

АТ-мансап бірнеше веб-сайттар туралы ақпарат береді, олардың кейбіреулері төменде көрсетілген.

1. <http://www.wetfeet.com/careers-industries/careers/information-technology>
2. http://www.cio.com/article/101314/The_Hottest_Jobs_In_Information_

Technology

3. <http://www.careeroverview.com/technology-careers.html>

4. <http://www.techcareers.com/>

5. <http://information-technology.careerbuilder.com/>

Үздік көзі үшін АТ-білім беру көмегімен табуға болады, арнайы топтың мүдделерін АТ-білім беру (SIGITE) <http://www.sigite.org/it-model-curriculum>.

Жалпы мәліметтер компьютерлік техника, бағдарламалық қамтамасыз ету және пайдаланушылар болуы мүмкін, кез келген саны мәтіндерді компьютерлік сауаттылық сияқты табылған.

1. Фуллера, Ф. және Ларсон, В. Компьютерлер: Технология түсініктемесі. Сент-Пол, Миннесота: ECM Paradigm Publishing, 2010.

2. Мейер, М., Бабур, Р. и Pfaffenberger, В. Болашақтағы компьютер. Жоғарғы Saddle River, Нью-Джерси: Prentice Hall, 2007.

3. Laberta, С. Компьютерлер сіздердің болашағыңыз. Жоғарғы Saddle River, Нью-Джерси: Prentice Hall, 2011.

4. Уильямс, Б. және Sawyer, С. Ақпараттық технологиямен қарастырылған. Нью-Йорк: Mc Graw-Hill, 2010.

5. Снайдер, Л. Беглость ақпараттық технологиямен: дағдылар, тұжырымдамалар және функционалдық мүмкіншілік. Жоғарғы Saddle River, Нью-Джерси: Prentice Hall, 2010.

6. Бекман, Г.және Бекман, Технологиялар Б. Ертең және сіз. Жоғарғы Saddle River, Нью-Джерси: Prentice Hall, 2008.

2. Компьютерлерді ұйымдастыру және жинақтамалар

Бұл тарауда компьютердің аппараттық құралдары, орталық процессор (БҮ), жад, кіші жүйесін енгізу/шығару, сондай-ақ құрсымды қарастырды. Әрбір осы компоненттердің сипатталған егжей-тегжейлі қамтамасыз ету үшін, оның рөлін түсіну керек. CPU қарау кезінде, мемлекет цикл-таңдау-орындау көрсетеді. Мысалы, иллюстрациялар кезеңін таңдауды орындау үшін бағдарламасы пайдаланылады. Процессор компоненттері өздері зерттеді: басқару блогы, арифметикалық, логикалық құрылғысы және регистрлер. Өткенде жад иерархиясы, жады жарияланған және әр түрлі конфи нысандары жад талқыланады: RAM, SRAM, DRAM, ROM, виртуалды жады. Талқылау кезінде құрылғыларын енгізу және шығару, адам-компьютер тақырыптарды өзара іс-қимыл (HCI) және эргономика параметрлерінің асты сызылады. Бұл тарау туралы мәселені жинауды қамтиды, әр түрлі компьютер компоненттері талқылайды. осы тараудың мақсаты, оның ерекшелігі, АТ-студенттің берік іргетасы үшін компьютерлік техниканы қамтамасыз ету.

Осы тараудың оқыту мақсаты болып табылады:

1. Компоненттердің және олардың рөлін анықтау.
2. Цикл-таңдау-орындауды сипаттаңыз.
3. Әсер ететін өнімділік процессор сипаттамаларын талқылаңыз.
4. Әр түрлі деңгейлерінің иерархиясын сипаттаңыз.
5. Әр түрлі типті құрылғылардың рөлін сипаттау, енгізу және шығару.
6. Бар адамның денесінде және HCI әсерінің маңыздылығын талқылаңыз, ол құрылғыларды енгізу / шығару.
- 7 Компьютердің құрамдас бөліктерінің бірін суреттеу.

Компьютерлік ұйымдастыру - зерттеу компоненттері, компьютер, олардың функциялары, олардың құрылымының (олар қалай жұмыс істейді) және олар қалай бір-бірімен байланысты болып табылады. Бұл

тақырып жалпы кіріс болып табылады.

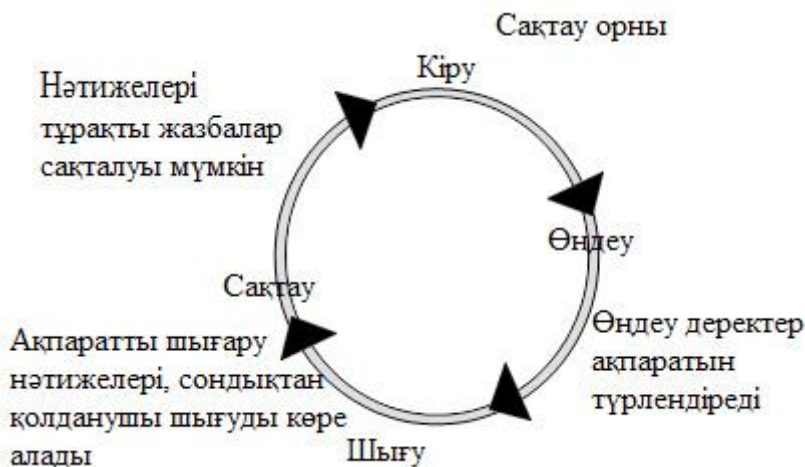
Пайдаланушыдан кіріс сигналын компьютермен не істеу керектігін хабарлайды (бағдарлама),

Мұнда біз бұл компоненттерді қарайтын боламыз, біз олардың функцияларын және олар қалай бір-бірімен байланысты екенін көпті көруге болады. Зерттей отырып есептеу ұйымдастыру, сіз түсінік алуға қалай компьютерлердің жұмыс істеуі мен оның маңыздылығын, әрбір негізгі компоненттерін 1-тарауда енгізілген. Бұл тарауда бөліміне арналған компьютерлік техника мен құрастыру процесі (ғимараттар) компьютер құрамдас бөліктерінің бірі аяқталады.

Компьютер мынадан тұрады: орталық процессор (CPU), жад және деректерді сақтау, сондай-ақ перифериялық құрылғылар (оның ішінде, сол немесе өзге де нысанда желісіне қосу). Компьютер төрт операцияны орындайды:

1. Кіру
2. Өңдеу
3. Шығу
4. Сақтау

IPOS цикл енгізу-өңдеу-шығару-сақтау-шамамен сипаттайды, біз компьютерді қалай пайдаланамыз? Біз кіріс мәліметтерді өңдеуге, нәтижелерді шығаруға және кез келген ақпаратты сақтауға, өмір бойы



2.1-сурет BOBX айналымы.

оны сақтауды қалаймыз.

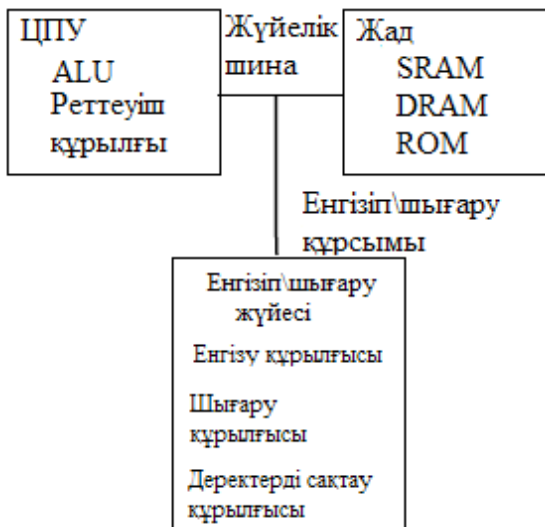
2.1-сурет ВИС айналымын суреттейді. Осы компьютерді суреттеу деңгейде шын мәнінде бізге жүрмейді екенін, мысалы қалайша жад сақтау айырмашылығы және өңдеу жүзеге асырылады. Осылайша, біз осыған көп назар аударамыз.

Компьютердің құрылымы

2.2 Суретте көптеген компьютерлік жүйелерінің жалпы құрылымы көрсетілген. Сіз төрт компонентті көре аласыз: орталық процессор (немесе процессор), жад, енгізу/шығару (I/O) жүйе және құрсым. Процессор құрылғыны білдіреді, ол тек өздерінің бағдарламалары мен командалар түрлі компоненттерін компьютерде орындайды. Жады әрбір бағдарлама қолданатын іске қосылған бағдарламаларды және деректерді сақтайды. Жүйені енгізу/шығаруды, барлық перифериялық құрылғыларды (кіру, шығу, сақтау, желі) сақтайды, болашақта ұзақ мерзімді камтиды. Деректер сақтау құрылғылары (қатты дискілер, таспа).

Заманауи компьютердің құрылымы

Құрсым құрылғысы, ақпаратты арасында жылжу үшін әрбір



2.2-сурет. Заманауи компьютердің құрылымы

компоненттің мүмкіндігін беретінін білдіреді.

Бұрын, біздің ойымызша, компьютердің компоненттерін, біз түсінуіміз керек, бұл құрылғылар компьютерде сандық құрылғыларды білдіреді. Компьютер ішінде сандық деректер түрінде ұсынылған электр тогының мәні бір, екі жай: жоғары немесе төмен. Жоғары елеулі ток ағатын арқылы компонент дегенді білдіреді және төмен дегенді жоқ немесе ток ағатын білдіреді. Біз нөмірі 1 жоғары деңгейдегі токты және 0 ток төмен болуы мүмкін (немесе жоқ). Сонда түсінік үшін шина жеңіл құрылғы болуы мүмкін. Құрсым бірнеше сымдардан тұрады, әрбір сым 1 бит (бір) 1 немесе 0 мүмкіндік береді, ол арқылы жоғарғы немесе төменгі ток) ағады. Біз құрылғылардың сыйымдылығын еске сақтаймыз. 1-тарауда (1.4-кесте) талқыладық, бірақ қазіргі уақытта, біз үш терминді анықтаймыз: аз (жеке 1 немесе 0), байт (8 бит пайдалана отырып, сегіз сымдар құрсымына, әдетте ең төменгі бірлік деректерді беру), сөз (бүгін, компьютерлер, 32-биттік немесе 64-биттік сөздер мөлшері, сөздер типтік деректердің мөлшері). Автобус шын мәнінде үш бөлімнен тұрады: құрсым мекен-жайы, құрсым, басқару құрсымы. Бұл үш бөлігінен келесі операциялар:

1. Құрсым мекеніне CPU жіберу үшін немесе жадтың ішкі жүйесін енгізу / шығару пайдаланылады. Үйренішті процессор тірек нүктесін қаламайды немесе орналасқан жерін процессор нөлдік нүктесі сақтап қалғысы келеді. Процессордан астам - MoNly жадқа қарағанда мекен-жайы болып табылады, I / O. Назар аударыңыз, бұл мәнінің шамасын пайдаланылуы тиіс немесе команда болуы мүмкін.

2. Құрсымды басқару процессор тарату үшін команданы пайдаланылады. "Оқу" немесе "жазу" жады, "кіру", "шығу", немесе "саған қолжетімді" құрылғысының енгізу / шығару командалары мүмкін. Құрсымды



2.3-сурет. Жүйелік құрсымның қосылу процессоры мен жады.

басқару, сондай-ақ түрлі құрылғыларды компьютерге жіберу кері сигналдар үшін CPU пайдаланылады. Бастапқы сигнал ретінде белгілі ұзу, ол сұрау құрылғы көмегін білдіреді (мысалы, дискжетек, принтер). 3. Құрсым деректері үшін (оның ішінде бағдарламалық команда) жад процессорын пайдаланылады немесе жад деректер процессорын, енгізу-шығару құрылғыларымен және процессор немесе жад болады.

Құрсым көлемі туралы деректер - әдетте, компьютердің сөздер мөлшері болып табылады. Мөлшері атаулы құрсым мөлшері негізінде мекен-жайы. Құрсым мөлшерін басқару негізделген саны әр түрлі команда тыс процессор жіберуі мүмкін. Сурет 2.3. құрсым деректері, бәлкім, 32 бит мөлшерінде, сондай-ақ мекен-жайы және автобус мөлшерін бақылау, бірқатар мәселелер түрі тәуелді болады. Бұл суретте автобус көрсетілген, онда жүйелік құрсымы болып табылады. Басқа автобус, жергілікті автобус компоненттерін CPU [ALU (арифметика /логикалық құрылғы) біріктіреді, басқару блогы, регистрлер] және тағы бір құрсым құрылғысын енгізу/шығаруды бірге біріктіреді.

CPU түсіну үшін, компьютерлік бағдарламаны біз түсінеміз. Бұл дегеніміз процессордың бағдарламаны орындауы. Осылайша, біз алдымен қарапайым бағдарламаны қарастырайық. Келесі бағдарлама C жазылған болса да, кез келген тілде жазылуы мүмкін дерлік. Жалғыз нәрсе синтаксисті өзгерту // белгілерді көрсетеді. Мына түсініктемелері бағдарламаны түсінуге көмектеседі.

Бұл бағдарлама екі бүтін есігін, мәндерін пайдаланушы, егер бірінші аз екінші, ол есептейді $c = a + b$ (сомасы екі) айтпесе, ол есептейді $c = a - b$ (минус b). Содан кейін оларды шығарады. Мысалы, қолданушы 5, 10, енгізеді ол есептейді $= 15$ және шығу 15. Қолданушы орнына 5 және 3 енгізеді, есептейді $= 2$ және шығу 2. Бұл өте маңызсыз, бірақ ол қызмет үшін, біздің мысал болады.

Кейін C бағдарламасына не жазылған? Менің бұл бағдарламаны іске қосуым керек. Жоқ. Бірде-бір компьютерге бағдарламалау тілін түсінбейді. Оның орнына арнайы бағдарлама деп аталатын құрастырушы іске қосамын. Құрастырушы бағдарламасын неғұрлым қарапайым тілінде аударады деп аталатын машина тілі. Тілдік нұсқасы машина бағдарламасының орындалуы мүмкін. Машиналық тіл архаистық болып табылады және әрі қарай өте қиын, сондықтан біз аралық нысаны бағдарламаның ассемблер тілін қарайтын боламыз. Әрине, кейін пікірлер болады // белгілері бар.

Тілдік нұсқа құрастыру дөрекі нұсқаулықтарға қарағанда С нұсқасы астам тұрады. Мысалы, бір scanf С операторы шын мәнінде, пайдаланылуы үшін мәндерінің уақытына кіруінің бөлінуі мүмкін және әр кіру жұп үшін сақтайды. С өтініші көптеген аз өтініштерді бөлінеді. Енді ассемблерге тиіс бағдарламасын талдайық, бағдарлама тілдің жоғарғы деңгейінде оңай болып келеді. Дегенмен, біздің бағдарламалар бір нәрсеге қол жеткізу үшін пайдалы, бізге әлдеқайда көп нұсқаулықтарды ассемблер тіліне қарағанда нұсқаулықтың тілі жоғары деңгейде. Сіз екі тілдер түрінде осылайша ойлайтын едіңіз: нұсқаулық ассемблер тіліндегі сомындар және бұрандама құрылысы, оқыту тілі пәнінен жоғары деңгейдегі дайын компоненттер сияқты топсалар бар. Компьютер істеу мүмкін емес нәрсе сияқты немесе басқа өтінішпен тікелей бір командада, бірақ ол жүктемені қосуды жасау мүмкін, дүкен немесе секіруге жекелеген операцияларға қатысты.

С сияқты бағдарлама, компьютер бағдарламасы ассемблер тілін орындай алмайды. Осылайша, біздің бағдарлама келесідей қайта құрылды, келген ассемблер тілі машина тілі болуы тиіс. Себебі, біз бағдарламаны ассемблерде көрсеттік, бұл түсініктемеге қарағанда, бағдарлама машиналық тілде оңайырақ жұмыс жасайды.

Ол бастапқы бағдарлама жазылған және жоғары деңгей тіліндегі сияқты С түрлендіреді, машина тілі компилятор бағдарламасын білдіреді. Ассемблердегі бұл бағдарлама ассемблер тілінен машина тілін аударады. Сонымен, ойлап көріңізші, біз алдыңғы машина тілі бағдарламасын құрастыруды ауыстырдық. Түпкілікті б бөлігі.

1000100	00000000000000000000100001	-input (C	пернетақта
лавиатура)			
1000111	0010011000100101101010001	-магазин	ИГД
1000100	00000000000000000000100001	-input (C	пернетақта
лавиатура)			
1000111	0010011000100101101010010	-магазин	ИГД

Қалған машиналық тілде бағдарлама бөлігі жоқ, өйткені онда ешқандай мағынасы жоқ еді, жоғары тізбесіне қарағанда 1s және 0s. Егер сіз 1s және 0s, немесе екілік сандарды қарасаңыз, бұл әрине, екі бөлікке форматталған. Бұл екі бөлігін әр топтың жұмыстарын білдіреді (алғашқы 7 бит) және операнд немесе тірек нүктесі. Мұнда операция бірін білдіреді, бәлкім, 100 түрлі нұсқаулар. 1000100 "кіру" ұсынады және 1000111"дүкенді" білдіреді. Сол жерде операциялары үшін"қосу", "SUBT", "JGE", "салмақ", және "секіру". Операнд жад

ұяшыққа жазылған екілік жүйені білдіреді (мысалы, деректер элементі орында сақталады 0010011000100101101010001, мекен-жай 5,000,017, ал мәні 000000000000000000100001 құрылғылар нөмірін білдіреді 33, пернетақта. Бұл тіл машиналық код мысал түрде тұрады, бұл тым ерекшеленеді, біз таба алар едік, егер біз зерттеген нақты машиналық тілдер бағдарламаны шығарса.

Бағдарламаны іске қосу

Жарайды, компьютерлік ұйымдастыруға оралайық. Біз бұл қарапайым бағдарламаны қосайық. Біріншіден, онда қосу болса, ол операциялық жүйе бағдарламасын жүктейді, сақталатын дискідегі еркін бөлігінде жад. Біз операциялық жүйені санауға және оның ұяшық жады 5000000. Себебі нұсқаулықтың 17 бағдарламасы сақталуы, жүйелі түрде жад ұяшық 5,000,000-нан 5,000,016 дейін. Айнымалы, а, b, c, санауға болады, кейін бағдарламада жад ұяшықтарында 5000017 арқылы 5,000,019 бірден сақталады, тиісінше, а, b және c. жүктегеннен кейін операциялық жүйе деп басқару процессор (CPU), жұмыс үшін осы бағдарламаны бастау керек. Енді біз егжей-тегжейлі, бұл процессор және жадыны істеуді түсіну керекпіз. Процессор тұрады: басқару блогы, АЛҚ және регистрлер. Осы тіркелімдер деп аталатын бағдарламалық есептеуіш немесе ДК-мен шатыстырмау үй компьютер). Дербес компьютердің мекен-жайы бірінші команданың бағдарламадағы жад 5000000 алады. Басқа регистр регистр командасы (IR) деп аталады. Ол ағымдағы нұсқаулықты сақтайды. Басқа тіркелім жалаулар жай-күйі деп атайды (SFS); ол ең соңғы нәтижесі ALU сақтайды. Есептеуіш тұрғысынмен қатар нәтижесі оң, теріс, нөл, толып кетуді, үзуді туғызды, мәні тіпті тепе-тендік және осылайша, бұдан әрі. Әрбір осы элементтердің 1 биті сақталады, сондықтан SF бірнеше нәтижелерін сақтауға болады, дегенмен көпшілігі биттері 0-ге тең болады. Басқа тіркелімдер, деректер тіркелімдер, олар деректерді сақтайды, біз қазіргі уақытта біз оны пайдаланамыз. Бір батарея (айнымалы ток), арнайы тіркелімі деректер деп аталады, ол сақтау үшін пайдаланылады. Бізге қарауға болса, процессор бағдарламасын іске қосады. Процессор, яғни белгілі бір таңдау-орындау кезеңін орындайды. Бұл кезең мәні болып табылады, процессор алдымен командасын алады, содан кейін орындайды. Шын мәнінде олар көп. Типтік іріктеу-орындау кезеңі (кезеңге байланысты процессорлар ерекшеленеді) төрт немесе бес, мүмкін көбірек кезеңдері болады (сурет 2.4).

1. Реттеуіш құрылғысы құрсымға бағытталып жадға «оқы» деген сигнал береді, жад IR-де жинақталу үшін деректер құрсымына нұсқаулықтарды қайта жібереді.
2. Реттеуіш құрылғысы IR-дегі нұсқаулықтарды анықтап таниды.
3. АК-дегі деректерді қолдана отырып ALU-дегі нұсқаулықтарды орындап АК-де нәтижелерді қайта қойылады.

Біздің мысал бес бөліктер кезеңі келесі кезеңдеріне санауға болады:

1. Жадтан нұсқаулық алу.
2. Нұсқаулық көмегімен машиналық тілі, микро кодты таратып жазу.
3. Тіркелімнен операндыны алу.
4. Командасын орындау.
5. Кері тіркелім нәтижесін сақтау.

2.4 суретте 3-5 тармақтары ретінде барлық көрсетілген 3-қадамды орындау, команданы пайдалана отырып, тіркелімнің деректері. Дегенмен, көптеген процессорларда бірнеше, тіпті ондаған тіркелімдер бар. Бұл үшін қадамдары: 3 және 5 бөлінуі 4-қадамды.

4-қадам біздің бес бөліктерінің командасы орындалады. Алайда, өздеріңіз көріп отырғандай, бұл жалғыз-қадам емес: қажетті ассемблер тіліндегі операцияларды орындау. Барлық кезеңдерінде бағдарлама дұрыс жұмыс істемейді. Микрокодта 2-тармағында аталған, бұдан әрі осы тарауда талқыланатын болады. Сонымен, алдыңғы бөлімде бағдарламасы талқыланды, орындалатын болады, бұл бес бөліктердің кезеңін қалай көрсек болады? Бірінші, процессор жүзеге асыру бөлу болып табылады, нұсқаулықтардың бірінші бағдарламасы. Бұл арқылы жіберу маңызы бар қала, сақталған ЖК (бірінші командадағы ұяшықты жад) жад бойынша адрестік құрсымына және белгі бойынша құрсымы "әрі қарай" жіберу. Естелік ретінде қабылдайды, мекен-жайын және " әрі қарай командасын " орындайды, оқу үшін осы ұяшықта жад қол жеткізуде. Не болса да орталық процессор бойынша деректер құрсымында табады, содан кейін үйге жіберіледі, бұл жағдайда, орталық процессор деректермен кері жіберіледі, бірақ бағдарлама бойынша нұсқаулықты жағдайда біздің бағдарлама бойынша бұл команда "Input" болып" екеулік түрінде жазылған: 1000100 + 00000000000000000000100001

Тек командасы орталық есептеуіш бөлім көмегімен алса, ол АТ болып сақталады. Осы кезеңде аяқтау үшін, процессор ӨК - 5,000,001 арттырады, енді орналасқан жері келесі нұсқаулар үшін алынатынын көрсетеді.

Келесі кезеңі нұсқаулықтарды жолатпау болып табылады. Басқару блогы үзіліп, нұсқаулық екі немесе одан да көп операция бөліктен тұрады (бұл жағдайда алғашқы 7 бит) және операнда (лар). Мәні бойынша, басқару блогы операцияларды табу кестесі үшін 1000100 командасы "Input" кенеседі. Бұл процессор орындау үшін осы нұсқаулықты, операцияны іске қосу құрылғысын орындауға тиіс, берілген мекенжай бойынша операндыны хабарлайды. Операнда екілік 33 сандармен (біздің жалған компьютер) пернетақта болып табылады.

Өйткені жоқ операнданы алу үшін, 3 кезең өткізіледі. 4 кезең командасын орындау болып табылады. Нұсқаулық енгізуді талап етеді, процессор деректермен алмасады, құрылғысымен іске қосылады (пернетақта) және келесі тірек нүктесін енгізеді. Орындау командасы енгізу болып табылатыны тән, бұл процессор жалғастырмайды, әзірге онда пайдаланушы енгізеді. Сол кезде, орталық процессор байланыстыру бойынша нүктесіне деректер құрсымына алынады және әкеледі, оның әр түрлі процессорлары бар, әр түрлі ұзындығы үшін олардың таңдау-орындау кезеңі, тек бірнеше ондаған БҰ. Бесінші және соңғы сатылап ауыстыру маңызы бар құрсым деректердің тіркелімі айнымалы ток болады. Енді біз бірінші команданың толық орындауын бағдарламада көрдік. Бұдан әрі не болады? Барлық таңдау-орындау кезең қайталанатын. Шын мәнінде, ол қайталауға жалғастырады, әзірге бағдарлама өз жұмысын аяқтайды.

Екінші команданың бірінші қадам таңдау керек команданы еске алу. Енді компьютер бар мекен-жайы 5,000,001, сондықтан процессор командасын бұл жерде жад алады. ӨК мәні құрсымның мекен-жайы орналастырылады, басқару блогы туралы есептеудегі жады арқылы құрсымды басқаруға белгі береді, сол мекен-жайды жады қарайды, мазмұн бойынша деректер құрсымына және элемент сақталған ИК қайтарады. Соңғы қадам таңдау фаза болып табылады, үстелуі ӨК - енді көрсетеді 5,000,002. Нұсқаулық АК"сақтау" болып табылады.

1000111 0010011000100101101010001

Сатысында қайта кодтау анықталады, бұл операция сақтауды талап ететін, орын ауыстыру тірек нүктесінің айнымалы кернеу тогы негізгі жады. Мекен-жайы алатын тірек нүктесін айнымалы ток желісі ретінде екінші бөлігінің нұсқаулықтары сақталады, 5000017 (мекен-жайы). Нұсқаулықты орындау үшін, ИК соңғы бөлігі ауысады "құрсақ мекен-жайы айнымалы ток желісінен деректер құрсағы ауысады, ал басқару

блогы "естелік жазу" туралы құрсағы белгі береді. Орындау командасы енді негізгі жад қолында, ол құрсақ деректерінің мәнін қабылдайтынын сақтайды және оны ұяшықты еске алады, мекен-жай құрсақ алынған (5,000,017). Бұл команда бесінші қадам ретінде процессор қажет емес, сондықтан сақтауға ештеңе талап етпейді.

Келесі екі команда ұқсас түрде кездеседі, бұл нұсқаулықты 5,000,002 және тиісінше 5,000,003, ал екінші байланыстыру нүкте МЕМ жерде сақталады 5,000,018 (б). Сол уақытқа дейін бұл екі команда орындалды, ЖК орналасқан 5,000,004 және 5,000,017 жад ұяшықтар және 5000018 алғашқы екі кіріс мәндерін сақтайды, тиісінше а, б.

Бесінші АТ командасы алынады және сақталады. Бұл нұсқаулық "жүктеу". Нұсқаулық мекен-жайын 5,000,017 атаулы оқу дабылы арқылы және жад белгісі оқулары бойынша жібереді. Жадыға қарап жоғарыға тірек нүктесі бұл жерде сақталады, оны қайтадан деректер құрсауы бойынша жібереді. Сақтауға алынған тірек нүктесін айнаымалы тогы нұсқаулықтың қорытындысы болып табылады.

Мынаған назар аударыңыз, бұл тараудың алдыңғдан айырмашылығы "оқу жадынан операциялар" сипатталған, бұл тұрақты есте сақтау құрылғысы шынайы нөлдік нүктесі емес нұсқаулар. Сурет 2.5 "жүктеме" (жоғарғы бөлігі сурет) арасындағы айырмашылықты және "дүкен" (төменде Рогдық сурет) нұсқаулықтардың оқу жадын және жазу жадын суреттейді.

Алтыншы нұсқаулық "SUBT" бесіншіге ұқсайды, "жүктеу" басталады. Командасы бұрынғыдай "Subt" таңдалады, ИК сияды және ДК бірлікке ұлғаяды. Басқару блогы содан кейін бұл нұсқаулықты кодтайды. Дегенмен, орындау кезеңі үшін жүктеме айырмашылығын шегеру, өйткені бұл команда екі бөлек қадамдарды орындауды талап етеді. Біріншіден, естеліктер оқылады, меніңше бұл жағдайда "жүктеу" мекен-жайы болып табылады 5000018 (ауыспалы б). Дегенмен нүкте жадынан емес байланыстыру қайтарылады, айнаымалы ток сақталады.

1. Айнаымалы АТ шешу үшін модульді құрсымы сигналдарды оқу жады арқылы басқару.
2. Жад мекен-жайын DATUM бойынша деректер құрсымына қол жеткізуін қайтарады.
3. Айнаымалы ток желісінен деректер сақталады.

1. АТ мекен-жай құрсымына дейін тірек нүктесінің айнаымалы ток желісі арқылы деректер құрсымы жазбаларын басқару блогы, жад сигналдарын басқару.

2. Жад дүкендердің мекен-жайларына Datum жылғы құрсым деректердің орналасуы жадына қол жеткізу.

Сурет 2.5 Оқулары жадыға (жоғарыда) байланысты жазу жады (төменде). Оның орнына мазмұнды айнымалы ток желісінен және нөлдік нүктесіне қайтып жадынан АЛУ жіберілді. Басқару блогы АЛУ үшін орындау шегерілуіне белгі береді. Шегеру сызбасы екі деректер мен нәтижесі, шығыстың бірі АЛУ болып табылады, оның мәні айнымалы ток желісінен минус жады (яғни А - В).

Дүкен нәтижесі: осы нұсқаулықтың бұл нәтижесі АЛУ және оны жылжыту айнымалы ток болып табылады.

АЛУ, сондай-ақ тудың жай-күйін белгілейді, негізінде нәтижені шегеру. ГЛС белгілеу үшін операциялардың нәтижелерін ALU пайдаланылады, мысалы, нәтижесі оң, нөлдік немесе теріс немесе туындаған, нәтижесінде толық немесе үзуді орындауға тиіс. Жалаулар, әдетте PF ретінде белгіленеді (оң туы), ZF (ту нөлден), NF (теріс туы), OF (ту толып кету), және осылайша, бұдан әрі. Кез келген АЛУ кем дегенде операциясы әкеледі, бір ту теру (мәні өзгеріп отырады 0-ден 1). Бұл жалаулар мынадай нұсқауларды пайдаланылатынын көреміз.

Жетінші нұсқаулық (5,000,006) бірінші нұсқаулық болып табылады, ассемблер арқылы оқу кодын ұғыну мүмкін емес. Ол былай дейді: "JGE-тағы". Бұл "артық секіру немесе басқа орынға тең". Орналасуы басқа орын. Өйткені, бірінші команда "жад бағдарламасы радио 5000000, онда "басқа" ереже орналасқан 5,000,011 (бұл 12-ші нұсқаулар). Алдыңғы бөлімде оралу бағдарламасы. Барлық алдыңғы нұсқаулықтар ретінде және бұл команда ("JGE-да) АТ жадынан алынады және қайта кодтайды. Орындау жұмыстардың кезеңі мынадай: Егер қандай да бір PF немесе ZF белгіленсе (азайту оң немесе нөлдік нәтижесін әкелді), содан кейін компьютердің орнына "тағы" алып тастау керек. Егер NF орнатылған болса, онда ештеңе істемейді.

Бұл дегеніміз, егер алдыңғы бұтақтар орнында "басқа" нұсқаулық (азайту) оң немесе нөлдік нәтижесін әкелсе, PF және ZF мұнда "артық немесе тең" командасы анықтайды. Егер команда "кем" болса, біз ғана NF пайдалана аламыз. Басқа да "тең", "тең емес", "артық" және "кем немесе тең" салыстыруды қамтиды. Соңында осы нұсқаулықтың компьютері 5,000,007 (келесі дәйекті нұсқаулық) немесе 5,000,011 (орналасуы басқа) болады. Сегізінші нұсқаулықтар, болғанына жетінші тәуелді. Немесе процессор 5,000,007 немесе 5,000,011 нұсқаулығын алады. Қандай да болса, мынадай үш нұсқаулықтың дерлігі бірдей. Олар "жүктеу", "қосу / Subt B" және "дүкен". Яғни, үш топ нұсқаулықтарды

орындамайды $c = a - B$ немесе $c = a - b$ қарай бір, екі жиынтығының коды орындалады. Біз нұсқаулықты бастауын болжап 5,000,007 орындадық, онда нұсқаулық бойынша 5,000,010 "секіру оқу" болып табылады. Бұл команданың мәні PC болуын өзгертеді 5000012 (жанында орналасуы). Қандай жол мұнда бізді келтіруі соңғы үш нұсқаулықта "жүктемені азайту", "шығу", "тоқта" командасы шығаруды қабылдайды, бұл айнымалы ток желісінен және құрылғыға шығару тізімін ауыстырады (2049, болжам бойынша, монитор). Процессор кезінде "Токта" командасын орындайды, операциялық жүйесі бағдарламасы аяқталады және басқару қайтарылады.

Орталық процессордың рөлі

Біздің бағдарламалар процессорларды өңдейді. Ол бұл құралды қамтамасыз етудің екі бөлігі арқылы жүзеге асырады. Олардың біріншісі АЛУ болып табылады, ол барлық арифметикалық және логикалық операцияларды орындайды. АЛУ жеке сұлбасын орындау үшін әр түрлі операциялары бар. Көбейткіш үшін пайдаланылады. Қалқа бөлу үшін пайдаланылады. Екі мәнді салыстыру компаратор үшін пайдаланылады (мысалы, анықтау үшін, $bar > b$). Басқа берілестерді ажыратып қосу сұлбаларын орындайды, бұрылу және жұптық есептеу (3-тарау қарау). Екінші жабдықтардың бір бөлігі басқару блогы болып табылады. Блокты басқару бақылау (командалық) барлық компьютер компоненттері үшін жауап береді. Мысалда бағдарламасының орындалуы, мұндай белгілерді басқару блогы жібереді, естелік ретінде оқу немесе жазу жады. Ол сондай-ақ АЛУ белгілерін орындау сияқты азайту немесе салыстыру жібереді. Басқару блогы кезең-таңдау-орындауды бақылайды. Біріншіден, ол іріктеу-кезеңін орындайды. Содан кейін ол кодсыздан нұсқаулық микрокодты түсіреді. Бұл өз кезегінде, блок басқару туралы командасын береді. Осылайша, басқару блогы болады, бөлінген жиынтығы команда жылдамдығын таңдау үшін команда кезеңі және әрбір машиналық команда. Кейбір нұсқаулықтың операсында талап етеді, сондықтан басқару блогы өңдейді, қалай операндты алуға болатынын көреміз (3 кезең таңдау-орындау кезеңі бағдарламасының орындалуын сипатталған). Кейбір нұсқаулықта ALU талап етеді, сондықтан басқару блогы хабардар етеді, АЛУ сызбасына тиісті басқаруға болады. Кейбір нұсқаулықтың нәтижені сақтауды талап етеді (5 кезең таңдау-орындау кезеңін орындау бағдарламалары), сондықтан басқару блогы тірек нүктесі көзден айнымалы ток жеткізеді (немесе басқа тіркелім).

Бұл нақты операциялар осы такте өтуге тиіс. Мысалы, тандау, бірінші қадам жылжыту компьютер жадына атаулы құрсымы үшін маңызды және сигнал оқуларын орындау үшін. Бұл екі команда ғана болып табылады, екі іс-әрекеттер орын алған күні осы тактің ішінде. Микрокод үшін бұл қадамға бинарлы тізбесі нақты компоненттерді алуға тиіс командасы болады. Бинарлы листинг толығымен нөлдерден тұрады, өйткені біз, тандау командасын көріп отырмыз, тек екі компонент компьютердің барлығы болады. Сондықтан микрокод үшін бұл қадамның екі 1s, ал қалғандары 0s болып табылады. Осылайша, сол машина тілінде микрокод көрінеді. Алайда, тіпті біздің тіл үлгісі машина бағдарламасында 32-биттік нұсқаулар тұрды, біздің нұсқаулықты микрокоды болып табылады, күнге дейін компоненттері компьютер ретінде бар. Мысалы, егер команда үшін 50 түрлі компоненттері нұсқаулықтың микро кодының ұзындығы 50 бит болады.

100000000000000100000000000000000000000000000000000000

Бір команданы микрокодты кейде микро-операция деп атайды. Әр түрлі микро-операциялар үшін әр қадамды тандау-орындау кезеңі, сондай-ақ бір немесе бірнеше микрофон операциялар үшін әр нұсқаулықтар машина тілінде болады. Естеріңізге сала кетейік, 2-қадам: машиналық командасын бірі-тандау-орындау кезеңін түрлендіру тиіс (мысалы, "жүктеу") микрокоды. Кейін бұл қадам ғана болып табылады, мәселе басқару блогын жіберу белгілерінде, онда әрбір бит микро-операцияларды білдіреді, жіберу арқылы басқа құрсымды басқару сызығы. Бұл тақырып өте жақсы және сізге түсінікті, қам жеменіз, өйткені осы мәтінде маңызды емес материал жоқ. Біз әр түрлі тіркелімдер туралы айтқанбыз. Екі түрлі сынып тіркелім: сол басқару блогын пайдаланылады және АЛУ үшін пайдаланылатын деректерді сақтау. Басқару блогы ӨК пайдаланады (есептеуіш), ИК (тіркелімінің командасы), SFs стек көрсеткіші, және мүмкін басқа да. Дегенмен біздің мысал бағдарламасы және бағдарламаның орындалу сілтемесі бірыңғай тіркелім деректер, айнымалы ток (батарея), қазіргі заманғы компьютерлер бірнеше, мүмкін жүздеген тіркелімдер. 8088 процессор Intel (ерте компьютерлерде IBM PC пайдаланылған), содан кейін Pentium процессорлар пайдаланады, төрт тіркелімінің бүтін сандық деректер

аттары EAX, EBX, ECX және EDX. Басқа да компьютерлер алар еді, R0, R1, R2 тіркелімдер деп айтуға болады. Бұл деректер тіркелімдерді уақытында сақтайды, АЛУ есептеулер үшін пайдаланады. Қазіргі заманғы компьютерлер тактық жиілігін бірнеше ГГц (гигагерц). Бұл термин шын мәнінде нені білдіреді? 1 гигагерц (ГГц) 1 миллиард тактық кезең секундына немесе сағат жылдамдықпен жұмыс істейді 1 миллиардтық секундты білдіреді. Бұл дыбыс 1-ГГц процессор 1 миллиард орындайды, нұсқаулық үшін сес ОНД. Бұл жалған. Естеріңізге сала кетейік, таңдау-орындау циклі бес кезеңнен тұруы мүмкін немесе одан да көп. Яғни, әр кезең таңдау-орындаудыкөрсетеді, кем дегенде 1 ырғақты кезең үшін бұл кезеңнің орындалуы тиіс. Қарастырайық, іріктеу кезеңі сипатталғандай, бағдарламаның орындалуы. Бұл кем дегенде үш ырғақты талап етеді.

Бірінші кезеңде ӨК мәні жөніндегі мекен-жад құрсымына жіберіледі және басқару блогы туралы есептеудегі жадыға белгі береді.

Екінші кезеңде жады бойынша командасы деректер құрсымына қайтарады.

Үшінші кезеңінде ИК нұсқаулық сақталады және ДК артады. Егер біз болжағандай, бір іріктеу кодтауды алады, егер операнд таңдау бір кезең аласа, орындау командасын бір орында және үш кезең, Stor ИНГ бір кезең нәтижесін алады, онда бір іріктелініп орындау кезеңі баламалы бір машинада оқыту тілі созылады, алты сегіз тактық кезең.

Байқасаңыз, әр түрлі процессорлар түрлі ұзындығын таңдау-орындау кезеңі болады. Процессор кезеңі 1 ГГц болуы мүмкін, бескезеңді талап ететін сегіз іріктеулерге арналған нұсқаулық. Процессор 2,5 ГГц болуы мүмкін 12-ші кезеңі талап ететін 20-ға жуық іріктеулер арналған нұсқаулық. Бұл жылдам ба? Процессор 2,5 ГГц-ден аса жылдам сағаттары, бірақ кезеңдердің ең көп саны орындау командасы үшін алады. Шын мәнінде, бұл жағдайда екі нұсқа бірдей өнімділік ($20 / 2,5 \text{ млрд} = 8 / 1 \text{ млрд}$).

Басқа тәсілі өлшеу жылдамдығы процессор үшін есептесе, команданың 1 секунд бойы орындалуы мүмкін. Бұл MIPS мағынасы жиі көрінеді (секундына миллион нұсқаулықтар). Өйткені компьютерлік сұлбасы және көптеген есептеулерді пайдалануды талап етеді, мәндердің өзгермелі нүктесі (сандар ондық нүктесі) тағы бір мүшесі мегафлопс (миллиондаған нүктелі флоат ИНГ операциялар секундына). Олардың алайда бірде-бір біздің тарихымызды жылдамдық процессорын айта алмайды. Процессордың жылдамдығы, сондай-ақ әсері мынадай:

1.Сөздер мөлшері, деректер тасымалдануы кез келген уақыт сәтінде өлшемін шектейді (мөлшерлі сөздер, әдетте деректер мөлшерінің

құрсымы мен тіркелімдер мөлшері; мөлшерлері аз сөздерден әдетте түрі көп деректер құрсымына және ұзақ мерзімді кезеңдерде орындау үшін үлкен мәндер есептеулер).

2. Өнімділік кэш (келесі бөлімді қарау) және жылдамдығы, жады мен мөлшері.

3. Бағдарламаның өзі (кейбір бағдарламасын ресурстар талап етеді, олар баяу немесе басқа артық уақыт).

4. Қандай компьютер жұмыс істейді, бірнеше бағдарламаларды бір мезгілде салыстырғандағы бос жүйесі.

5. Операциялық жүйенің өнімділікке әсері.

6. Виртуалды жад өнімділігінің әсері.

7. Көптеген басқа да сұрақтар.

Алу үшін нақты процессор өнімділігінің көрінісі, компьютерлік сәулетшілер сынай процессорлар эталондық бағдарламаларға қарсы болады. Өнімділік қандай да бір бағдарламаны адастырып енгізе алады, сондықтан пайдаланылатын тесттер, тексеру үшін люкс процессорлар, әр түрлі көріністері процессорды тексереді. Біз жақсы түсіну үшін жалпы өнімділігі эталондық процессорды орындайды. Бірақ бұл жерде ГГц процессор маңызды болып табылады, алданған рейтинг мүмкін емес, ол сізге бір нәрсе процессоры туралы, бірақ көп емес, сіз ойлануға болады. Егер компьютерлер сіздің ұйымыңызға тапсырысырылса, сіз жақсы қызмет көрсету ғана емес, әрі қарай орауды және ГГц рейтингін көріп және әрбір процессор жарияланымдар өнімділігін оқу (мысалы, Consumer Reports, PC World) және веб-сайттар (www.cpubenchmark.net, www.geek.com).

Жадтың рөлі

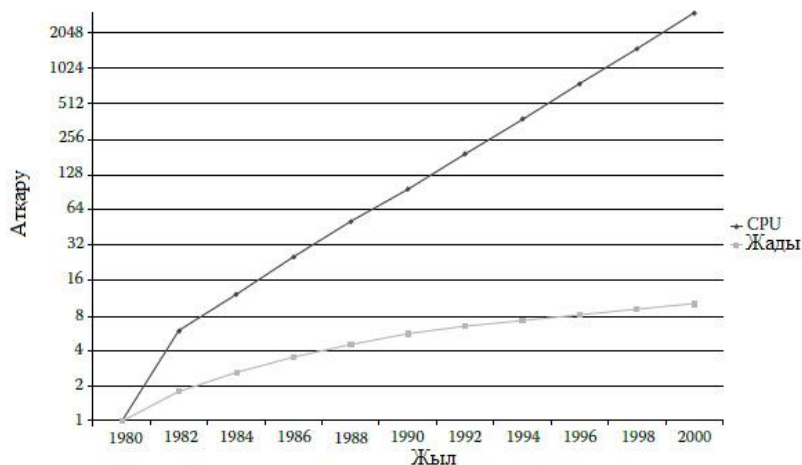
Компьютерлер тарихының басында, жад тек деректерді есте сақтайды деген түсінік қалыптасты. Компьютерлік бағдарлама орындалады, бір команда бір рет перфокарталарды енгізеді. Дегенмен, компьютермен сияқты ENIAC (электрондық сандық интегратор және компьютер), олар орындауға қабілетті 5000 операциялар секундына енгізгенде, команда бағдарламаның перфокарталарды өнімділік ENIAC оқырман перфокарталарды азайтады. Джон фон Нейман бірінші бағдарламасының компьютері сақталатын. Идея мынада: компьютер жадысында бағдарламалық код ретінде және деректер сақтауға болады. Алайда, айтарлықтай жад мәселесі (RAM) және оның процессор жылдамдығы әлі де айтарлықтай жыл сайын жоғарылайды, бірақ қол жеткізу ram болады, бірте-бірте көптеген жылдар бойы жақсарылады.

Сурет 2.6 үлкен жылдамдықта процессорды салыстырғанда орташа уақытты жақсартуға қол жеткізуін суреттейді. Мынаған назар аударыңыз, бұл өнімділігін ұлғайту Y осі бейнеленеді, орналасқан 2 бөліктерінде экспоненциалды шкаласын көбейтеді. Мысалы, 1982 жылдан 1986 жылға дейін қол жеткізу жады әрең өсті, ал өнімділік процессор төрт есе. Өнімділік процессор шамамен әрбір екі жыл сайын екі еселенеді, ал негізгі уақытта қол жеткізу көптеген жылдар бойы жад әрең өседі, негізгі қол жеткізу жылдамдығы жады артта жылдамдығы процессор көбірек келеді.

Процессор бағаланған жад, кем дегенде, бір рет нұсқаулық-таңдау командасын еске алу. Кейбір нұсқаулықтың оқулары жадтан (команда жүктеу) талап етеді.

Нәтижелеріне естелік жазу (нұсқаулықтың дүкені). Шын мәнінде, кейбір процессорлар әрі қарай бірнеше операциялар мен жазуға, суретке әрбір команда мүмкіндік береді.

Жалғастырмас бұрын, RAM анықтаймыз. ОЗУ жедел есте сақтау құрылғысын білдіреді. Идея мынада: біз ақпаратты блокты еске алу командасын оқуға жібереміз. Жады жоғарығы элементіне қарайды және оны кері қайтарады. Екінші жағынан, біз жады блогындағы мекен-жайға жібереміз, тірек нүкте командасын жазу. Содан кейін бұл тармақта есте сақтайды. "Кездейсоқ" термині мекен-жайына жіберіледі, кездейсоқ азайтылады. Бұл термин сондай-ақ кез-келген мекен-жайға қол жеткізуді қамтуы тиіс, бұл тек уақытқа қарамастан, қандай мекен-



2.6 сурет. Жадқа ену уақыты кезінде процессордың жылдамдығын жақсарту.

жайға жіберілді. Бұл жедел жады түрлері ерекшеленеді, әр түрлі уақыт саны орналасуына байланысты-тармағының іздестірілуде сақталады (мысалы, таспалар қажет болуы мүмкін, көптеген кері айналдыруы жылдам және диск көп уақыт үшін / басының жазуы жылжуға тиісінше саналады). Өкінішке орай, RAM термині бірнеше мағыналы емес, үш нысанды жады, барлық түрде ОЗУ-ды бағалау.

Біз бұл нысанды RAM ретінде динамикалық жедел жады (DRAM) және статикалық ОЗУ (SRAM) және ТЕСҚ. ПЗУ тек оқу үшін қолданылатын жады. Бұл жадта үнемі толық ақпарат сақталады, сондықтан ол тек оқу үшін емес болуы мүмкін, жазылады. Көптеген компьютерлер бағдарламасын жүктеуді сақтауға ПЗУ негізгі қатар жүйесін енгізу / шығару (BIOS), бірақ сәл басқа болады. Біз туралы бұл мәтінде дискіде алаңдатты. Екі басқа да нысандарын гат, DRAM үлкені болып табылады. Біз миллиардтаған бір чипіне ДОЗАНЫ конденсатор кішкентай дәрежеде тұруы мүмкін екенін айтамыз. DRAM да өте арзан. DRAM, алайда, CPU салыстырғанда баяу болып табылады (Сурет 2.6 қарау). Осылайша, бір жағынан біз арзан құралды қамтамасыз ету үлкен жадты сақтауды, бірақ екінші жағынан, процессорға қол жеткізу әлдеқайда баяу болады (коэффициентімен 25-ден 100-ге дейін!).

SRAM салынғанды пайдалана отырып, бірнеше транзисторларды, триггерді блоктар деп аталатын SRAM құру үшін екі регистрлерді CPU және кэш (оқылады "ақша") жады. Кэш жаңа технологиялармен есептеулерде қарағанда, DRAM өте бағалы болып табылады. SRAM әлдеқайда көп DRAM-ға қарағанда, сондықтан біз әдетте, әлдеқайда оның біздің компьютерлерде аз пайдаланамыз. Дегенмен, SRAM да дәл солай тез, орталық процессор. сондықтан процессор қол жеткізе алады,



Сурет 2.7 Жадтың иерархиялары.

SRAM үшін нұсқаулықтар және мәліметтер, процессор еместі күтуге тура келеді, бұл кезде DRAM қол жетімді жад болып табылады.

Бізге қызықты шешпек боламыз. Өйткені бізде аздаған SRAM бар, біз кепілдік береміз, қашан бізге бір нәрсе еске ауысады (көшіріледі) SRAM жадынан DRAM? Бұл үшін компьютер иерархиясы деп атаймыз. Ол бірнеше қабатты қамтиды, әрбір қабаты баяу жүргізілуде, кем қымбат, бірақ әлдеқайда үлкен ыдыстар төмен болады (Сурет 2.7).

Иерархия жад регистрлері бөлігі процессор болып табылады. Онда бірнеше ондаған тіркелімдерге дейін, олардың әрқайсысы қабілетті сөз сақтауға, сондықтан бірнеше ондаған процессор болуы мүмкін.

Кейбір процессорларда екі тіркелімдер, қабілетті сақтауға екі сөздер бар.

Бірнеше жүздеген байт жад тіркелімі. Олар шамамен жүйелік тактілі сигналдар сияқты жылдамдықпен жұмыс істейді. Екі түрлі жиі деңгейдегі кэш-жад бар, ол чиптен тыс. Кэш кристалда орналасқан процессоры, дегенмен, сол облысы КПУ ретінде тіркелімдер. Дегенмен кэш чипіне сол технологияны, тіркелемді, косымша адрестеу аппараттық құралдарды қажет тудырады, кэш чипі сәл баяу болуы мүмкін, екі есе баяу болмауына қол жеткізу қажет. Кэш офф-чип орналасқан жеке аналық тақта технологиясы кэш чипінде пайдаланады. Дегенмен де, көп дискі тұру жылғы процессор, ол уақытты талап етсекэштен тыс чипті болып табылады, 3 кэш чипінен 10 есе баяу. Өйткені кэш кристалда орналасқан процессорлар, ол бірлесе отырып, қалған бөлігінде процессор бейнеленетін. Кэш-офф-чип әлдеқайда көп болуы мүмкін және әлдеқайда жад көлемі көбірек. Кэш чипіне тек 32 немесе 64 КБ тұруы мүмкін, онда кэш офф-чип 1-ден 8 МБ болуы мүмкін. Тіпті бұл жағдайда, кэштен тыс чиптің әлдеқайда DRAM қуаттарда аз болып табылатын, кем дегенде 4 ГБ. Дегенмен, жылдамдығы DRAM, әлдеқайда баяу, әлі күнге дейін мүмкін 25-тен 100 есе баяу процессор болады.

Назар аударыңыз, жад иерархиясы DRAM-мен аяқталады. Сондай-ақ, әдетте, біз бір мезгілде бірнеше бағдарламаларды іске қосамыз. Жөндеу қажеттілігі үшін жадқа қарағанда үлкен саны, біз негізгі жадыда кеңейтудеміз. Виртуалды тұжырымдамасын пайдалана отырып, виртуалды жадқа қарамастан, қазіргі уақытта пайдаланылады, белгілі кеңістік ретінде есептеледі.

Кеңістік соты алдын-ала қатты дискідегі жеке айдау сақталады, белгілі бір бөлімі ретінде есептеледі. Бұл бөлім айырмашылығы қалған файлдық жүйесін дискілік кеңістік құрылды, ол мүмкін-жылдам бөлінуі, тіпті

егер бөлу болып табылады, соншалықты кеңістік тұрғысынан тиімді пайдалану. Бірақ бола тұра, бұл тез әрекет етуге, біз ауыстыруға кей бөліктерін еске алу қажетпіз.

Әр деңгей еске алу ретінде "кері жүріс" иерархиясы келесі жоғары деңгейінде қарастырылған. Біз жүктеу арқылы бүкіл бағдарлама үшін кеңістік деректерді дискіден бастаймыз (немесе алмалы-салмалы тасығыштарды) виртуалды жады мен көшіру ғана бөлігін (мысалы, бағдарламаны іске қосу) DRAM жады дереу қажет. Сонда, біз нұсқаулықтың қатынау және деректерқалай бастаймыз, олар дәптерлерінің көшірмелері құпиялық чипіне жасалады. Кейін кэш чипіне кез келген қосымша элементтері толтырылады, тиелген " кэш бірінші қажет нәрсені алып тастаңыз.

Ілгерілетуге қарай жоғары иерархия шектеу бойынша кеңістік көп, сондықтан лақтыру жағдайлары жиі кездеседі. Таңдау ақылға әкеледі элементтердің кэшінде процессор қажет. Таңдау сапасының нашарлауына әкеледі. Біз есте кэш-жад жылдамдығы бойынша түсу тиімділігі (жиі болса, біз кэшты табамыз). Таңқаларлық, бірақ тіпті шағын кэш чипіне, хит мөлшерлемесі 98%, 99% - ға дейін жетуі мүмкін. Жылжыту иерархиясы бойынша төменге қарай, ұрып қарқынын жақсарту. Бұл вариация өнімділігін арттыру үшін неғұрлым қымбат.

Енгізу және шығару рөлі

Жүйені енгізуден/шығарудан тұрады, кіріс және шығыс құрылғыларының (құрылғыларды қоса алғанда, деректерді сақтау), құрсымды қосу үшін құрылғыларды бір-біріне жүйелі құрсымды, сондай-ақ интерфейстік құрылғылар сияқты карталар мен порттарды екпіндеу. Қатаң айтқанда, компьютер енгізу немесе шығару талап етілмейді, құрылғылары жұмыс істей бастады. Компьютерлік бағдарлама деректерімен бірге бағдарлама мен нәтижелері жадында сақталады. Дегенмен, енгізу және шығару құрылғыларының, біз емес, компьютердің жай-күйі, өзара іс-қимылына байланысты. Компьютерде нәтижелерін көруді біз айта алмаймыз. Ең ерте нысандарын енгізу және шығару шектелуі магнитті таспалар, перфокарталарды және компьютерлік басып шығару. Ешқандай тікелей өзара іс-қимыл жоқ; оның орнына, пайдаланушы дайындау сияқты бағдарламалық коды мен деректерді пайдалана отырып, телетайп құрылысы, оның нәтижелері арналған перфокарталар пайда болады (бір картаға сәйкес нұсқаулықтар немесе нөлдік нүктесі). Стек перфокарталарды кіру арқылы есептеуші перфокарталарды және сақталады магниттік таспада болады.

Жаңалықтар орнатылады, деректерді енгізеді және бағдарламалық қамтамасыз етуді орындайды, ал кері магниттік таспада шығу сақталады. Жаңалықтар жойылатын болады және басып шығару принтері, мәліметтері орнатылған. Бұл өте қанағаттанарлық түрде компьютерді пайдалану!

Бүгін, әрине, құрылғылардың үлкен саны қол жетімді жасау үшін компьютерге кез келген жағдайлардың саны қол жетімді. Кілт сөзі I/O бүгін қарапайым болып табылады. Біз есте дизайн философиясын бұл өзара іс-қимыл, адам-компьютер (HCI). "HCI, біз компьютерді пайдалану ретінде адам-бағдарланған емес, машина-бағдарланған.

2.1-кестеде тізімделген кейбір құрылғылар жиі енгізу / шығару пайдаланылатын бүгін және негізгі себебі оларды пайдалану. Сақтау құрылғылары-деректерді кестеден алынып тасталды, бірақ ішкі және сыртқы қатты дискті қамтиды, оптикалық диск, флэш-жады, магнитті таспаға, есте сақтау құрылғыларының желісі. Сақтау желісі, әдетте, қатты диск, желісі арқылы қол жетімді."HCI, негізгі міндет-жәрдемдесу астам табиғи қарым-қатынас әдістері, компьютер. Бұл зерттеу аймағы, ол информатика салаларында әкеледі, психология, дизайн және денсаулық, атап айтқанда, құрамында ұсыныстар үшін қол жетімді компьютерді пайдалану. Мысалы, енгізу құрылғылары, Брайль қол жетімді пайдаланушылар үшін көру қабілеті бұзылған. Үлкен мониторлар және операциялық жүйелер, олар оңай өзгертуге экран ажыратымдылығы (мөлшері экрандағы нысандарды) сондай-ақ көру қабілеті нашарларға көмектесе алады. Микрофондар жиі адамдармен пайдаланылады пернетақта және / немесе тышқан пайдалана алмайды. HCI фокусталады құрылғыларда азайтуға мүмкіндік беретін, салмақты адам ағзасы арқылы шамадан тыс компьютерлік өзара іс-қимыл. Эргономика жобалау жүйелері мен объектілерін зерттеу болып табылады, жақсы сәйкес келуін адам пайдаланады. Компьютерлер үшін, олар мыналарды қамтиды, жақсартылған пернетақта қоятын, аз салмақты білезікті адам, жақсы көрсететін құрылғылар сияқты тышқан жақсартылған.

2.1-Кесте Енгізу/шығару құрылғылары

Бар кодты оқитын құрал	Бастапқыда қорларда пайдалынылатын ақпаратты жинақтайды
Камера	Бейнекөріністі (тұрақты немесе қозғаламы) енгізеді

Дисплей көзiлдiрiгi	Бейнекөрiнiстердi көруге арналған, виртуалды шынайылықта қолданылады
Сiлтеуiш	Көбiнесе компьютерлiк ойындарда қолданыылатын, iс-қимыл жасау үшiн/ақпаратты белiгеу үшiн пайдаланылады.
Микрофон	Дыбыстық ақпарат жазу үшiн, пернетақтану қолдануға қабiлетi болмағанда немесе ауқымды ақпаратты енгiзу керек болғанда пайдаланылады.
Монитор	Сыртқы құрылғы
Тiнтуiр	GUI терезелiк жүйеде пайдаланушының әрекеттерiн енгiзу үшiн пайдаланылады.
MIDI құрылғы	Музыкалық ақпаратты енгiзу үшiн пайдаланылады.
Графикалық планшет	Жазылған ақпаратты енгiзу үшiн, пернетақта қолайсыз кейде қажетсiз болған уақытта қолданылады.
Принтер	Тексттi немесе суреттердi қағазға шығару үшiн пайдаланылады.
Сканер	Қағаздан жазбаларды немесе суреттердi енгiзу үшiн пайдаланылады.
Диктофон	Музыканы немесе дыбысты шығару үшiн пайдаланылады.
Сенсорлық панель	Тiнтуiр қолайсыз болған уақытта қосымша тiнту құрылғысы
Сенсорлық экран	Тiнтуiр немесе сенсорлық панель қолайсыз болған уақытта саусақпен тiнту, нұсқаулықтар енгiзуге арналған қосымша енгiзу құрылғысы
Шарлық манипулятор	Тiнтуiр қолайсыз болған кезде (тасымалдау немесе денсаулық жағдайына байланысты) қосымша енгiзу құрылғысы

Екi пернетақта және пайдалану тышқан көптеген адамдардың синдромының дамуына әкелдi, қосымша арнаның бiрi-көптеген нысандарын қайталанатын жарақат стресс. Жиi қайталанатын жарақат орындаудан мұндай стресс туындайды, iс-әрекеттердi қайта-қайта тозуын тудырады. Мысалы, пернетақтаны пайдаланып дұрыс бұлшық қатайту мүмкiн алған зақымымен күрестi, бiлезiк үңгiртау синдромына

әкеледі, қашан компьютер алдында бірнеше сағат ішінде отыру болса, басқа да стресс нысандары жаман түрге әкелуі мүмкін.

Заңына оналту туралы 1973 жылғы рұқсат беретін гранттар мемлекеттердің жарнамалайтын қызметтер азаматтар үшін ауыр кемшіліктері бар, қолжетімділігін арттыру мақсатында барлық нысандарында АТ өзгерістер енгізілді (508). Белгіленген нормалар бөлімінде 508 бойынша АТ-өнімдердің ұсыныстарын қамтиды, оның ішінде, веб-сайттар, бағдарламалық қамтамасыз ету, операциялық жүйелер және аппараттық қамтамасыз ету. Мысалы, барлық үстелге қойылатын және портативті компьютерлер енді керек.

Заңына оналту туралы 1973 жылғы рұқсат беретін гранттар мемлекеттердің жарнамалайтын қызметтер азаматтар үшін ауыр кемшіліктері бар, өзгерістер енгізілді (508) қолжетімділігін арттыру мақсатында барлық АТ нысандарында. Белгіленген нормалар бөлімінде АТ-өнімдері бойынша 508 ұсыныстар қамтиды, оның ішінде, веб-сайттар, бағдарламалық қамтамасыз ету, операциялық жүйелер және аппараттық қамтамасыз ету. Мысалы, барлық үстелге қойылатын және портативті компьютерлер енді бастап слоттарды келуге тиіс, кеңейту қолдайтын түрлі НСІ құрылғылар (мысалы, микрофон, трекбол). Операциялық жүйелер қажет, рұқсат ету үшін өзгерту рұқсаты, экранды орналастыру үшін қабілеті веб-сайттар, қолдайтын үкіметі өтімділігі оңай болуы тиіс, мысалы, шешкенде пайдалану пернелер табуляция (немесе басқа ыстық пернетақта, тышқан, жылжуға да бір басқа.

Басқа секциялар 508 актінің оналтуы, көптеген басқа желілердің қолжетімділігін басу. Оларға ұлттық оқу материалдары стандарттар қолжетімділігін насихатталған Американдық қорсоқыр, UDL 2.0 басшылығының Ұлттық Орталығының әмбебап дизайн, оқыту үшін Web Content Accessibility Guidelines ұсынымдарына сәйкес Wide Web Consortium World. Олардың әрқайсысы түрлі технологиялар фокусталады, бірақ барлық олар идеологиясын ілгері етеді, қолжетімділігі маңызды болып табылады. НСІ арқылы, ақпараттық технологиялар неғұрлым ыңғайлы пайдалану үшін үлкен популяциясы.

Қолжетімділігі және жарақаттану деңгейін төмендетуге қарамастан, НСІ мақсаттары маңызды болып табылады, зерттеу бүгінгі таңда, ең алдымен, зерттеу, адам орталығы өзара іс-қимыл компьютерді қою. Деректер тұрғысынан адам қабілеті енгізуге қалай жақсартуға болады. Мысалы, микрофон ұштастыра отырып сөйленісті технологиясына айырыптану мүмкіндік береді, адамға енгізуге анағұрлым нақты, әрі жылдам қарағанда, пернетақта. Алу үшін өңделген ақпаратты (шығу)

болып табылады, монитор үздік құралмен шығару керек. 3-D көзілдірік қамтамасыз етеді және нақты ақпаратты беру тәсілі.

Мәселелер HCI жақсартылған құралдарды қамтиды, бірақ жетілдіру қолданыстағы құрылғылар. Екі мысал, қамтамасыз ету үшін неғұрлым жоғары дисплей жасау үшін шығу неғұрлым анық және қамтамасыз ету үшін артықтығы шығу, кепілдік беру үшін, демалыс белгілер түсінікті. Басқа тұжырымдамасын қамтиды тиісті ментальды модельдер адамзат ақыл ұсына отырып тиісті ақпаратты тікелей жақын орналасқан басқа да ақпаратты пайдаланумен байланысты визуалды шығару емес, мәтін үшін жақсы болуы мүмкіндігін есте адам, және, осылайша, бұдан әрі.

Бір соңғы сұрақ HCI, адамдық алшақтау стационарлық компьютер және онымен есептеулер алуға шешеді. Бұл тұжырымдама, оны кейде, кеңінен тараған компьютерлік технологиялары деп атайды, қазіргі уақытта қол жетімді арқылы портативті құрылғылар. Дегенмен, шағын экрандар, меніңше, жоғарыда аталған ұғымдарды қолдайды. Осылайша, жаңа құрылғыны енгізу / шығару қазіргі кезде әзірленуде, олар есептеулерді бізбен бірге жүргізуге мүмкіндік береді. Бұл құрылғылар сымсыз және ұялы телефон технологиясының кейбір неғұрлым озық кіріс және шығыс құрылғылар сияқты көзілдірік үшін дисплейлер, микрофондарды қосу үшін, құлақ аспап дыбыс шығару үшін. Оларды кейде алып жүретін бұл құрылғылар деп атайды, бәлкім, Мысал ретінде тасуды жасауға алмады, елестетіңіз, сіз автобуста жүріп келе жатырсыз. Құрсымсыз интернетті қамтамасыз етеді. Сіздің тасымалды құрылғылар үйде компьютер арқылы сымсыз қосылады, сондықтан барлық өңдеу және сақтау сіздің үй компьютеріңізде. Сіздің көзілдірік көрнекі ақпаратты мәтін ретінде елестетеміз, негізінде таспасы немесе бейнесі бар болады. Құлаққап қызмет ғана емес, сізге аудио шығуы, бірақ тырысып окшаулауға дыбыстар шығатын сыртқы әлем. Сіздің микрофонға айтуға мүмкіндік береді, компьютерге бақылау үшін сіз не істейсіз? Және, ақыр соңында, толығымен таза емес көзілдірік болып табылады, олар жартылай таза, сондықтан сіз сондай-ақ сыртқы әлем қажеттілігіне қарай көруге (мысалы, сондықтан сіз көре аласыз, сіз автобус аялдамасына келесіз). Бірақ көзілдірік жасауға ғана емес, сіздің компьютер елестету бейнесі. Компьютер, блоктау үшін жағымсыз бейнелерді болжауда оңды болды. Және де сіз автобуспен серуендейсіз, сіз жарнамалық хабарландыруларды көрмейсіз, автобустың орнына көрінетін жағында, осы суреттер ауыстырылады ақ дақтар немесе суреттерді таба тартымды. Ұйғарылған қандай дыбыс? Мүмкін, бұл бүгін, бірақ бұл емес, HCI жалғастыруда жақсара түспек.

Қатар алып жүретін, бар бірқатар басқа да нысандарын жаңа

технологиялар айта кету керек. Сенсорлық экрандар, әлбетте, көп бөлігі біздің бүгінгі өмірімізде, өйткені олардың бір бөлігі болып табылады және барлық смартфондар мен планшеттік құрылғылар. Сенсорлық экрандар құрылғылары енгізу және шығару болып табылады, өйткені олар дисплей бар датчиктер атап жерлерде адам ж / е контакт экраны. Сенсорлық экрандар болған 1960-шы жылдардың аяғынан, бірақ есептеуіш құрылғылар пайдалану үшін тым қымбат. Дегенмен, бүгінгі таңда, сенсорлық экрандар пайдаланылады емес, тек қалта құрылғыларын қолданады.

Жақын арада, сондай-ақ стандартты интерфейс үшін үстелге қойылатын және портативті компьютерлер табуға болады. Заманауи сенсорлық экрандар интерфейс қалам негізінде (бүріккіш) және әрекет етуге қабілетті бірнеше бір мезгілде жанау талап етпейді.

Виртуалды шындық (VR) технологиясын білдіреді, әлі де қалыптасу кезеңінде тұр, ол дүниеге келген, шамамен 1980-х. VR, компьютер жасайды иллюзиялық әлем арқылы әр түрлі түрлері, енгізу және шығару құрылғыларының, адам бұл әлемде тұнады. Адам өзара іс-қимыл жасайды, виртуалды әлеммен негізінде қозғалыс және бағдарлау бас арқылы гарнитура датчигімен қозғалыс және қолғаптар. Қолданушы көреді және естиді, виртуалды әлем арқылы көзілдірік және құлаққап (Сурет 2.8). Сонымен қатар, толық костюм мүмкін пайдаланушыға басқа да түрлерімен сезіну.

VR мен, адам жерде тиелуі мүмкін, ол әдетте, жету мүмкін емес сияқты



2.8 сурет виртуалды шындық гарнитурасы және деректер қолғабын қолдануда. (NASA рұқсатымен, [HTTP: //](http://))

жаяу Марс немесе жорық әрекеттегі жанарту немесе мұхит теңізінде жүзу. Тәжірибелік пайдалану VR барлау қамтиды, әзірлеу және талдау (елестетіп тексеру жүргізу ING ұшақ тек әзірлеу сатысында), білім беру және ойын-сауық. Соған қарамастан, әлеуеті, VR ұсынады, әдеттегі құбылыс жоқ, өйткені көптеген кедергілер, олар еңсеру керек. Бірінші кезекте шектеулер құрылғыларда енгізу / шығару үшін кескіндеу реалистік қалдыру Blanc көрсететін қандай тілеу көп жақсы. Сонымен қатар, ТЖ шамадан тыс санын талап етеді. Есептеу техникасы дұрыс модельдеу және иллюзиялық әлемді ұсынуға. Осылайша, ВР әлі өте қымбат қолғаппен деректер және гарнитураның құны \$ 500-ден \$ 5000.

Бір соңғы нысаны жаңа технологиялар енгізу / шығару айта кетерлігі желісі датчиктерді білеміз. Датчиктер құрылғыны білдіреді, мүмкін сезінуге біршама аспект сияқты қоршаған орта температурасы, қысымы, су, немесе дыбыс діріл. Көптеген құрылғылар, құрамында құрылғысы, әдетте, белгілі бір құрылыс орындарда шектелген тетігі орналасқан. Желілік құрылғысы бөліп таратылған жиынтығы тетіктер білдіреді, сондықтан құрылғы алуға мүмкіндігі бар неғұрлым шынайы "түрі" қоршаған ортаға. Бүгін тетік айтарлықтай арзан, сенсорлық желі неғұрлым кең таралған құбылыс. Із біріктіруге датчиктер сымсыз байланыс жасау үшін сымсыз сенсорлық желінің тұратын бүтін оннан бірнеше мың жеке датчиктер, олардың әрқайсысы мүмкін өзара іс-қимыл бір-бірімен және базалық компьютер деректерді өңдеу мағынасы үшін. Сенсорлық желі бар әр түрлі тұтыну құрылғылар, көлік құралдары, аспаптар денсаулық жағдайын бақылауды, сондай-ақ әскери аппараттық құралдары. Тұтас тетіктерден тұруы мүмкін барлығы бірнеше жүздеген доллар. Қосымшалар үшін сенсорлық желілерін қамтиды, бақылау, мониторинг атмосфералық жағдайлар (ауа райы, парниктік газдар, ауаның ластануы), денсаулық жағдайын бақылау, сейсмикалық белсенділік және кен орындарын барлау.

Сонымен қатар, орап талқылау құрылғыларды енгізу / шығару, біз сондай-ақ қарауға тиіс екі басқа, олар жақсартты мүмкіндіктері пайдаланушыларды қосу үшін өз компьютер құрылғылары. Бұл плагин ойнауға және USB. Идея плагин ойнауға, сіз жаңа құрылғыны қосу компьютерге кез келген уақытта бұл құрылғылар бір распознаваться операциялық жүйесімен, пайдалануға дайын. Алдында плагин ойнауға, сіз қайта жүктеу компьютер, сіз қосылған жаңа құрылғы болатындай құрылғы болуы мүмкін деп танылса, кезінде тиеу процесін. Plug-мен-ойын алғаш рет операциялық жүйелер Windows 95, бірақ, содан бері стандартында көптеген операциялық жүйелер.

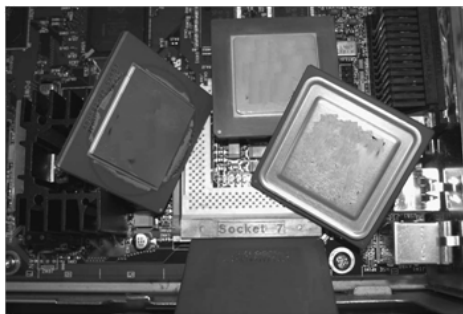
Ең көп таралған (емес, ғана) қосылу жаңа құрылғыны енгізу / шығару компьютерге USB порт арқылы. USB білдіреді ретінде Universal Serial Bus, стандартты интерфейсі бар компьютерлер үшін бастап 1990-шы жылдардың. Стандарт USB анықтайды байланысын көрсетеді, түрі, пайдаланылатын кабель байланыс тәсілі құрылғы мен компьютер арасында, сондай-ақ, коректендіру блогі , ол өмір сүре компьютер мен құрылғы арасында. Бүгінгі күні көптеген құрылғылардың болуы мүмкін қосылған USB арқылы қоса алғанда, пернетақта көрсететін құрылғылар-принтерлер, сыртқы дискілік жинақтағыштар, смартфон және басқа да ықшам құрылғылар. USB-диск немесе флэш-жад, пайдаланушыға береді, лайықты саны сақтаудың сыртқы құрылғыларын (1-8 Гб), бұл шағын болып табылады. USB порты тізбектелген және параллель порттар пайдаланылатын осындай құрылғылармен алмастырды. Сонымен қатар, порт USB құрылғы қуаттануды бере алады.

Құралдар және компьютерлерді жинақтау (монтаждау)

Бұл тарауда жалпы астам компоненттерін қарастыру және базалық құрастыру (монтаждау) үшін терезе негізінде үстелдік компьютерледі қарастырамыз.

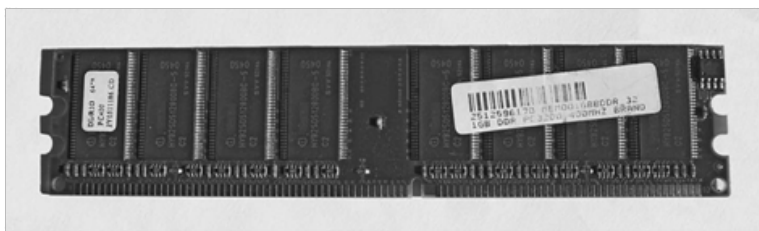
Орталық есептеуіш бөлім

Орталық процессор немесе процессор бір чипіне ретінде белгілі микропроцессорда сақталады. Процессоры бар бірқатар шағын штифттер бір жағында болады. Жақсы, ешқашан жанасуға осы қаданы бұғу, оларды, ең алдымен, процессор құртады. Сіз кірістіру процессор арнайы аналық тақта жалғағышы. Сіз байқағандай бір бұрыштарының процессоры бар ондағы үшбұрыш. Бұл орналастыру ұясы процессорға дұрыс қоюға көмектеседі.



Жады

Көпшілігі компьютер жадысында ram сақталады. Әдетте, ол-жады немесе ram, басқада түрлері жад компьютерде (ПЗУ, кэш). Жад микро сұлбасы қазірдің өзінде министрліктің баспа тақтасына келеді. Сіз ба төлемге арнайы слот аналық тақтадан тайып кетуіңіз мүмкін. Байқасаңыз, суретте осы сұлбаларымен байланыстары бар, олар өз бойындағы төменгі жағындағы. Бір жинағын басқаға қарағанда байланыстарды; бұл дұрыс бағдарлауды слот жадын шешуге көмектеседі.



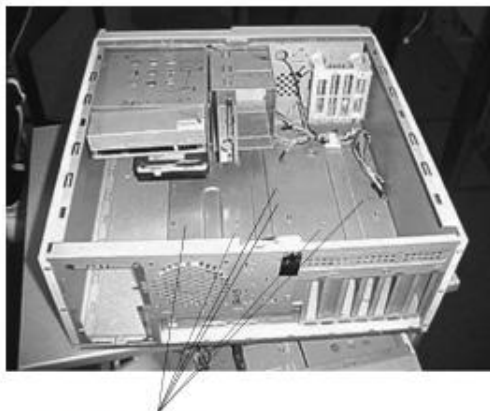
(Ласло Салаи, [http://commons.wikimedia.org/wiki/ Файл:DDR_RAM-3.jpg](http://commons.wikimedia.org/wiki/Файл:DDR_RAM-3.jpg).)

Жүйелік құрылғы

Бұл жағдайда, онда сіз бірнеше негізгі компьютер компоненттерін орналастырасыз.

Ең маңызды компоненті аналық тақта болып келеді. Аналық тақта шыңында оның бетінен "тұйығын" көтеру үшін отырады. Бұл шын мәнінде орналастырылуы бойында бір тараптың жүйелік блогы, ал төменгі кесіктен тұрады. Жүйелік блогында көптеген саңылаулар үшін бұрандалар көруге болады; олардың кейбіреулері кергіш болады. Сондай-ақ, әрбір салынған желдеткіш жүйелік блок болып саналады, сондай-ақ, кейбір құрылғылар деректерді сақтау сияқты оптикалық дискжелік және қатты диск болады. Мүмкіндігі бар басқа құрылғыларды ойық бойымен алдыңғы бөлігінде жүйелік блок орналасады.

Бұрандалы тесіктер үшін тұйықтар

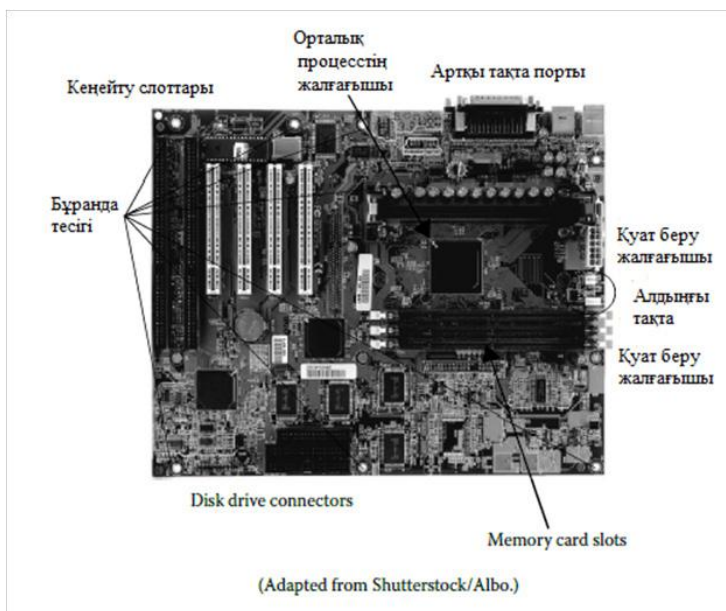


Бұрандалы тесіктер үшін тұйықтар

(Норман Роджерстің суретімен ұсынылған, [http://commons.wikimedia.org/wiki/Файл: Stripped-компьютер-case.JPG](http://commons.wikimedia.org/wiki/Файл:Stripped-компьютер-case.JPG).)

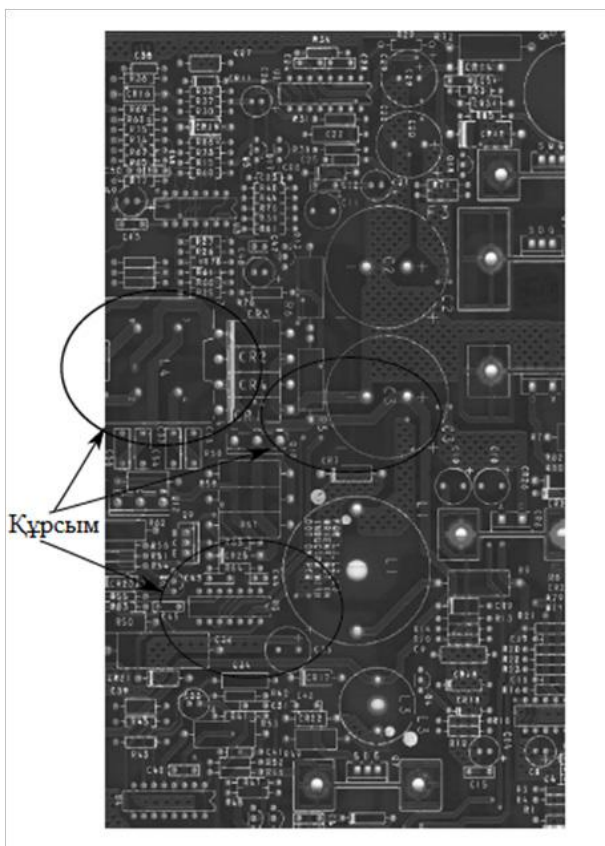
Аналық тақта

Аналық тақта арасындағы процессор арасындағы және де басқа да компоненттер арасындағы байланысты қамтамасыз етеді. Бұл құрылғылар аналық тақта арқылы слоттарды кеңейту және артқы порттарға қосылады. Аналық тақта бір бөлігін білдіреді және шыны талшықтан немесе пластиктен жасалған, онда розеткалар және автобус желілері бекітіледі. Процессордың ұяшықтарын ажырату үшін DRAM жады ұяларын қамтиды (немесе одан да көп), слоттары кеңейту карталары үшін, олар интерфейстерден бастап перифериялық құрылғылармен жабдықталған болып табылады, ажыратқыштар қосу үшін құрылғылар деректерді сақтау, қосу, қоректену, сондай-ақ порттары, олар ұстану арқылы артқы қапакты жүйесінің блогы болып келеді. Микросызбалар ROM жады қазірдің өзінде аналық тақтада бекітіледі.



Жүйелі тақтаға бар тесіктер үшін бұрандалар, сондықтан сіз оны жүйелік блокқа тіркеңіз. Аталған жүйелік блок, әдетте тік тұр, аналық тақта шын мәнінде орналасуы тігінен, сондай-ақ, бұл маңызды, аналық болады сенімді кабелі қосылуы. Сонымен қатар, тұйықтар көзі аналық тақта физикалық емес, ішкі бетінің жүйелі блогына қатысты. Себебі электр тогы арқылы ана ақылы ағады, егер металды түртіп ішкі беті жүйелік блок болса, бұл мүмкін кілттеу ток ағыны тиісті чиптер болмауы нәтижесінен болады.

Төменгі жағы ана ақысының құрамында "сым" -soldered желісін құрайтын бір бөлігі жүйелік құрылымдар бар. Осы желілерден де ток ағады. Сондықтан бұл сызықтар ештеңе істемейді, металлға тиісуге болмайды. Бұл себептерінің бірі, ана ақы шыңында "тұйық" болады. Бұл жүйелі ақы жылғы бүйір жағынан жүйелік блок мүмкіндігін арттыруға береді. Мынаған назар аударыңыз, бұл төменгі бөлігі ана ақысы өрескел (бұдыр) болып табылады, шағын үзімі аппараттық және дәнекер қылтиып тұр. Кезде сіз " ана ақысында жақсы орналастыру, оның жоғарғы бөлігі жұмсақ, берілетін бетінде ешқандай электр болмады. Қорап, аналық тақта кірді және полиэтилен пакеті көздеген аналық тақта болады.



(Courtesy of Shutterstock/jakelv7500.)

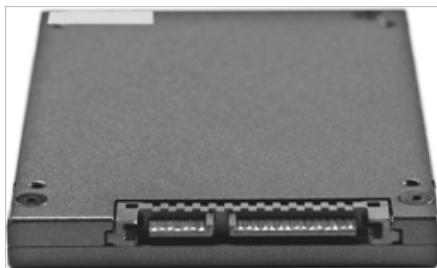
Қатты дискідегі жинақтаушы

Қатты диск стандартты бірлігі ұзақ мерзімді сақтау болып табылады. Алғашқы компьютерлер күндері, пайдаланылған магниттік таспа (шарғылар - орауышта содан кейін кассеталар магнитті таспалар және картридждер).

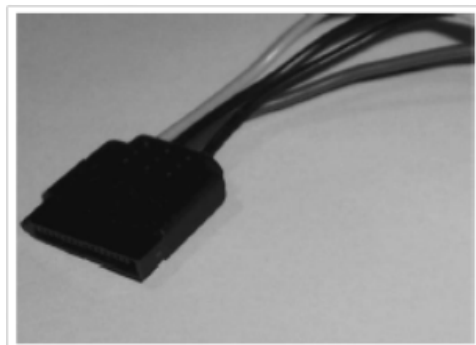


Берілген Hubert Berberich, HTTP: // Жалпы
wikimedia.org / вики / Файл: Seagate-ST4702N-03.jpg).

Бүгін, қатты диск үлкен сандарды сақтайды (мөлшері сақтау үшін шамамен 1/2 ТБ дейін толық ТБтән болып табылады). Біздің компьютерлер, сондай-ақ құрамында оптикалық дискілер көбірек мүмкіндік беру үшін біздің компьютерлерге қызмет ретінде ойын-сауық консоль (ойнату үшін музыкалық компакт-дискілерді және DVD фильмдердің) мақсаты үшін сақтау. Диск құрамында бірнеше пластина қатты дискте магнитті зарядтар түрінде ақпарат сақталады. Бастардың оқу/жазу қозғалу сәті үстінен, әр түрлі беттерден дисктер бар. Диск құрамында, сондай-ақ оймен механизмдер үшін дискілердің айналу және орын басының тұтқаны оқу / жазуды ауыстыру. Бүгінде екі типті дискілер бар: IDE (Integrated Drive Electronics), SATA (Serial Advanced Technology Attachment). Төменде SATA артқы дискі, онда сіз екі қосылыстарды көре аласыз, біреуі қоректендіру үшін және біреуі деректер үшін. Оң жағында күштік жетек SATA кабель үшін болып табылады. Назар аударыңыз, онда ажыратқыш SATA "L-тәрізді нысаны" оған көмектесетін шарлау ұясы бар.



(Courtesy of Shutterstock/Matee Nuserm.)



(Martixer алынған, HTTP: // жалпы

wikimedia.org / вики / Файл: Molex-к-SATA- power_009.jpg).

Артқы IDE дискінің үлгісі төменде көрсетілді. Оң жағында орналасқан ұяшығына IDE деректер. IDE Сымын жалғау үшін жетегінің тұрады төрт тесіктер сай болу үшін төрт тісшесінің сол жағында (көрсетілгендей мұнда жетек-аяқ емес, төрт тісшесі оң жақта болады) жетек орналасқан.



Қосымша аналық тақтаға және жетек құрылғыларға, сондай-ақ кірістіру қоректендіру блогына жүйелік блогын қосуға болады. Қоректендіру блогы үлкен санды кабельдерге қосылады. Бұл кабельдер қоректендіру үшін аналық тақтаға, процессорге, дисплей панеліне (жүйелік блоктың алдыңғы панелінде), желдеткішке және дискілерге қосылады. Бұл суретте кабельдер барлық жиынтығымен бірге көрсетілді. Сіз абай болу керексіз, кезінде әр түрлі қоректену кабельдерін, олардың орны аналық тақтаға немесе дискілерге қалай қосылуы, бұл кабельдер ыңғайсыз қосу үшін мүмкін дұрыс, осылайша олар сіздің жолыңызды алады, сіз

құрастыру процесін жалғастырыңыз.

Компьютерді сатып алу

Сонымен сіз компьютер сатып алғыңыз келеді. Таңдауыңыз қандай болады? Бұл сұраққа жауап берудің алдында, ол компьютерды қандай мақсатпен пайдаланатыңызды шешіп алыңыз. Бұл сізге қандай түрі керек екенін саралауға көмегін тигізер:

- Сервер
- Жұмыс үстелі
- Ноутбук
- Блокнотты компьютер

Егер сіз компьютердың түрін тандасаңыз, сізді тағы бір таңдау күтіп тұр. Егер сіз серверден басқа түрін іздесеңіз, сіздің таңдауыңыз осыларға түсер :

- Макинтош
- Windows операциялық жүйесінің басқаруындағы ДК
- Linux операциялық жүйесінің басқаруындағы ДК
- Басқалар (бұл негізінде ноутбуктер мен серверлер)

Енді осы таңдаудың негізінде, келесіні таңдауыңыз керек:

- Процессордың түрі (жылдамдығы, кэштің көлемі, 32-бит немесе 64-бит)
- Жедел жадының көлемі (DRAM)
- Қатқыл диск көлемі
- Оптикалық жинақтағыш?
- Монитор өлшемі
- Модемнің жылдамдығы (немесе желілік адаптер)
- Басқа сыртқы құрылғылар

Сіздің таңдауыңыз компьютерды қандай мақсатпен қолданатыңызда, қалаған бағдарламалық жасақтама талаптарына және қаражатыңызға қарай құрылады.



(Courtesy of Winhistory, <http://commons.wikimedia.org/wiki/File:At-netzteil.jpg>.)

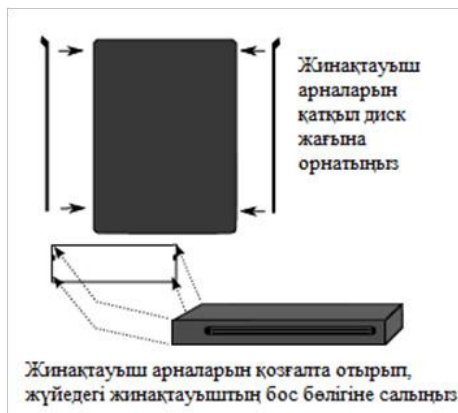
Сіз пернетақта үшін жалғағыш тышқан және дисплей туралы сұрай аласыз ба? Бұл қосылыстар аналық тақта болып табылады. Сіз панельдің артқы бөлігінде жүйелік блокты жойыңыз, әртүрлі жалғағыштары (порттары) қол сырттан жетімді. Енді сіз көрген әртүрлі компоненттерді қарастырайық. Бірінші, бұл барлық компоненттерді анықтап істеу керек. Олар ана ақысын, процессорды, жадты (чиптер), оптикалық бұрандаманы, сымбілезікті және құралдарын қосу керек. Дегенмен, сіз сымбелбеудікі керексіз имеу, дегенмен, электроникаға жақындау алдында оны қойғаныңыз жөн (аналық тақта, жаңқа). Одан әрі қорғаудың аппараттық құралдарын, кілемдік жабу жұмыс істемейді және айырылған киімді, статистикалық зарядты сақтап алып жүру. Кез келген чиптердің, неғұрлым әсіресе, CPU статикалық заряды зақымдауы мүмкін. Сонымен қатар, жұмыс орнын тазалықта, тәртіпсіздік және қоқысты тастауды сақтау.



(Courtesy of <http://www.freeimagespot.com/Countries/grounding-strap.html>, author unknown.)

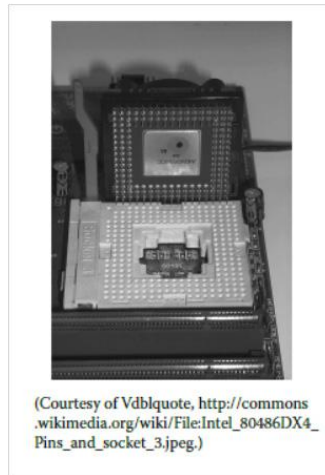
Кейін сіздің компоненттеріңізді ұйымдастыруды анықтау және бірінші қадам орнату қорек көзінің бөлігі азық-жүйелік блок болып табылады. Тастай бүйірлік жылғы жүйелік блок. Қореккөзі тиісті жерде және қауіпсіз, оны бұрандалармен қойыңыз. Сіз сондай-ақ қажетті артқы қақпағын порттардан алып тастау. Бұл тікелей жеткіліксіз.

Бұдан әрі, сіз DVD және қатты дискті кірістіріңіз. Сіз алдыңғы панелді жойыңыз, сіз жүйелік блок жерде DVD-дискті кірістірмеңіз (бұл үшін қатты дискінің қажет емес, сондықтан бетінде қатты диск жоқ болу керек екені көрініп тұр, бірақ DVD диск ашуы тиіс). Кейін, сіздің дискілер орнында болғанымен, оларға әлі ешқандай кабельдерді жалғамаңыз. Кабельдерге кедергі жасайтын басқа да іс-шаралар болады, сондықтан қосылыстардың соңғы қадамын жақсы сақтаған дұрыс.



Енді сіз компоненттерді аналық тақтада жинауды бастаңыз. Бұл үшін сізге сым білезік кию керек. Сіз тиісті жүктеме жерге соңына тұйықтау ременді металл жұмсауға, көбінесе металл "жасушалары" жүйелік блок. Пайдалана отырып, ана ақылын анықтау, тұйықтар жүйелік блогында қажет. Ана төлем тіркелу қажет емес, ол әлі келген жоқ, бірақ квинт "олардың дұрыс қалыпта тіректерге (солар арқылы ана ақысына нықтау үшін, онда сіз іздеу үшін бұрандалы аналық тақтада тесіктер сәйкес келсе, онда бұранда үшін тесік жағына бірлік жүйесін) өтуіне мүмкіндік береді.

Бірінші компонент енгізу үшін арналған ана төлем процессор болып табылады.

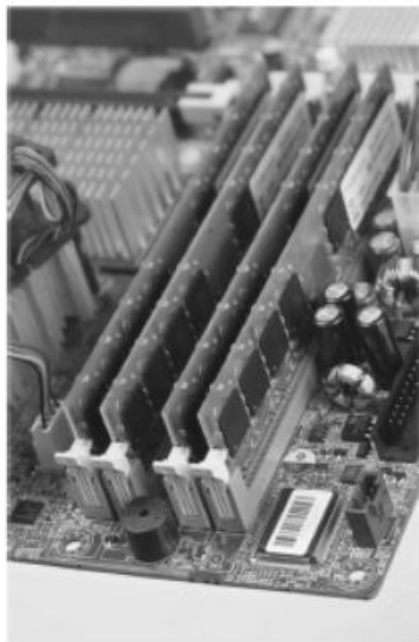


Сіз процессор үйіндісінің (көрсеткенге назар бөліңіз), иін тірегі бар құлыптау көтеріңіз, қайыrmалы қақпағы бар тік процессор, тайып кету процессор ұясы, қақпағын түсіріңіз, содан кейін тетікті түсіріп алғаныңызға көз жеткізіңіз.



Бұдан әрі сіз орталық процессордың суыту блогын орнатыңыз . Біріншіден, сізге жағынды жылу тарататын пасталар процессордың сыртқы бетіне орналастыру қажет. Сомасын сіз қойыңыз "процессор, өлшемі шағын кішкентай болады. Бұдан әрі, блок салқындатуды бағыттауға, сондықтан оны төрт қосқышы (қысқыш) процессорлық сокеті төрт тесік айналасында орналасқан. Бағытталған құрылғыны тексеріңіз, сондықтан, электр сия сымы оңай" қоректенуі ұя ана ақысы болады. Кейін ол төменде әрбір бір-бір төрт үшкір темірлерді дұрыс

орнатады. Кейін барлық төрт түйреуіштер арқылы ана төлемді растау үшін ана ақыны көтеріңіз. Содан кейін әрқайсысы үшін көрсетілгендей үшкір темірлерді бетінің шрифтерін қолданамыз.

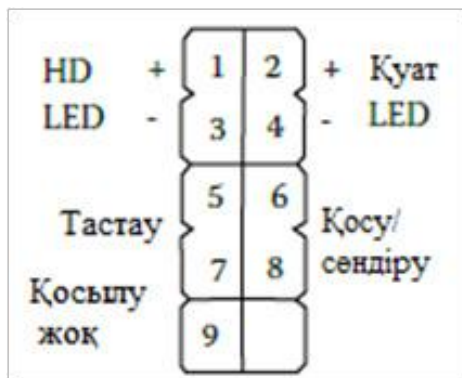


(Courtesy of Shutterstock/Jultud.)

Содан кейін, баспа ақыны еске алу және кірістіру, оның бірі слоттарды қарастырыңыз. Төлем үйіндісінің тиісінше иша саңылауының "оның саңылауы бар штифтпен салыстыру "балапан ұяда тексеріңіз. Көз жеткізіңіз шығыңқы жерлері бойымен тараптар паздары "ашық" болып табылады. Баспа төлемге арналған тесікті салыңыз, содан кейін пайдалана отырып, өз саусақтарын ғана түймесін төмен басып, қойындылары автоматты түрде жабылатынын көресіз. Сіз бұл қадамды әрбір сұлба үшін жад, сіздің компьютерге қосуды қайталаңыз.

Осы құрылғылардың орнында, сіз енді бұранда аналық тақтаны орын ағынсыз бұрын офф қою көмегімен орнатуды аяқтау үшін, сізге енді зэк қажет, арасындағы түстік белгілер блок қорегі, дискілер, желдеткіш және аналық тақта беріледі. Біріншіден, кабелі бар 2-байланыс төрт қосқышын табу (ескерту: қарай, қорек болғы, бұл бір ұясы сегіз-

байланыс мүмкін). Салымдар фразасында аналық тақтаны құжатта орнату керек. Мысалы, қосылыстар қоректендіру үшін ұстайды LED, түйме тастау төмендегі суретті көрсетеді (қайта жүктеу), қуат индикаторы мен қосу/өшіру. Ақау қосу жалғағышқа тиісті шрифтпен әкеледі, бұл компоненттер жұмыс істемейді. Диаграммадан төмен, 1/3 байланыс үшін қатты диск LED қолданылады. Pins 2/4 қуаты on / off (өшірулі) LED. Pins 5/7 үшін пайдаланылады түймесін қайтадан қосу. Pins 6/8 жарық қосу / өшіру үшін пайдаланылады.



Содан кейін, сіз дискілерді қосыңыз. Екі дискіні қосу үшін, оптикалық диск және қатты дискті алу керек. Ең алдымен, IDE sata қатты диск болып табылады, көп таралған компьютерлер PC стильде. Оптикалық жетегі SATA немесе IDE болуы мүмкін. Типін анықтау үшін, ол пайдалануға қосылады. Біріншіден, кірістіру дискілер бухт жүйелік блогында қолжетімді. Кірістіру оптикалық әкелу шыңында өзара іс-қимыл сыртқы әлеммен оптикалық жетегі болады, сондықтан ол жиі кездеседі, оның жоғарғы бөлігінде жүйелік блок пайда болады. Сіз осы бухтаның қапқағының үстінен оптикалық диск сыртынан қосасыз. Әр түрлі жүйелік әр түрлі типті бухта блоктарда болады. Дегенмен, барлығы істеу керек, бұны дисковод бөлікке салыңыз, содан кейін кілт жағындағы шығанағын бұраңыз.

Кейінекі (немесе) дискілерді өз бухталарында қосу керек, олардың аналық тақтасы және қорек блогын іске қосуды ұмытпаған абзал. Табу қуат кабелі қосылу үшін қолайлы, жұмыс қорек көзінің және тиісті деректер кабелі қосылу үшін ана ақысының көмегі зор. Соңғы біріктіру бұл қосу кабелі желдеткіштің қоректендіру кабелін – аналық тақтаға қосу. Қазіргі уақытта, сіз қосуға қоректендіру блогы - компьютердің құрылғылары және асқын кернеуден қорғау. Желілік сүзгі қосылған розеткаға қосуына

және енгізілуіне көз жеткізіңіз. Енді өз компьютеріңізді жүктеңіз!

Егер компьютер сәтті жүктелсе, BIOS ROM жүктеледі, өйткені дискіде ешқандай операциялық жүйе жоқ, ол терезеге жетпейді. Егер сіз компьютерді табысты жинасаңыз, онда Windows орнату компакт-дискін салыңыз, оптикалық диск жетегі компьютерді қайта іске қосыңыз. компьютер BIOS ROM жүктегеннен кейін, Windows орнатуға болады. Құттықтаймыз!

3. Есептеудің екілік жүйесі

2-тарауда біз цифрлық компьютерді жылжыту арқылы электр тоғы арқылы оның құрамдас бөліктері жұмыс істейді. Содан кейін қайта құрылғысын табу үшін ақпарат тәсілі екілік болып ұсынылады. Содан кейін, біз екілік файл көмегімен қалай есептеуге болатынын қарастырамыз. Бұл тарауда біз екілік жүйесін нөмірлеуін зерттейміз. Тарауда қайта құру екілік санау жүйесінің әдістері екілік үшін жазылмаған (оң) сандар, теріс бүтін санын, фракциялық және сандардың өзгермелі нүктесі. Бұл тарауда, сондай-ақ, сегіздік және он алтылық форматында жүйесін нөмірлеуді енгізеді. Таңба беруді талқылайды. Содан кейін осы тарауда бинарлыққа (Boolean) операцияларды, оларды есептеу. Ақырында, мемлекет үш мысалда екілікті енгізеді: желілік адрестеу, файлдарды, суреттерді, сондай-ақ шындықты табу үшін және қателерді түзету.

Осы тараудағы оқыту мақсаты төмендегілер болып табылады:

1. Қамтамасыз ету әдістері сандық түрлендіру арасындағы негіз: ондық, екілік, сегіздік және он алтылық үшін оң және теріс сандар және сандардың өзгермелі нүктесі.
2. Рәміздер компьютер жадында қалай сақталатынын сипаттаңыз.
3. Көрсету қолдану бинарлық (Boolean) операциялар AND, OR, NOT және XOR арналған екілік сандар.
4. Екілік кодты компьютерде назар аударып пайдалану, IP-мекен-жайлары, файлдарды сақтау үшін суреттер мен бит шындықты суреттейді.

Компьютер өңдейді және ақпарат түрінде екілік кодты сақтайды. Осылайша, біз екілік файл ретінде ұсынуға қарағанда ақпаратты екілік жүйеде қалай екілік (Boolean) операторларды қолдануға түсіндіреміз. Бұл тарауда екілік нөмірлеу жүйесі және бірнеше оған байланысты жүйелерді нөмірлеу. Бұған қарамастан, бұл ғалым бинарлық, ақпараттық тех-

нологияларды (АТ) түсіндіреді.

Нөмірлеу жүйесін, сондай-ақ белгілі санау жүйесі ретінде хат білдіру сандар жүйесі болып табылады. Математика жүйелерінің нөмірлеу саны шексіз, бірақ, ондық және екілік жүйесі тек екі үлкен дәрежеде шоғырланған. 3.1 кестеде ең жиі пайдаланылатын және тарихи пайдаланылатын жүйесі нөмірлеуді ұсынған.

Біз формальды жүйесін нөмірлеу негіздері ретінде анықтаймыз (оң санмен).

1. Кез келген саны, осы күні сандар 0-ден – 1 қамтитын болады.

2. Интерпретация сандар болып табылады, оның * қуат мәні.

3. Бағанада күші негізделеді немесе ұстанымын билік = $\text{base}^{\text{column}}$, онда мақала түсуінен бастап оңнан солға қарай отырып есептеледі, шеткі оң жақ бағанасында -баған болып табылады. Мысалы, ондық базасы болып табылады 10. Бұл біз пайдаланатын 0-ден 9-ға дейін сандар. Күш 10^{column} . Егер біздің нөмірі 130 болса, онда $1 * 3 + 102 * 101 + 0 * 100$ мәні ұсынады немесе 1 бағанасында жүздеген, 3-бағанасында ондаған, және 0-бағанасында.

Мысалы, ондық базасы болып табылады 10. Біз пайдалануға болатын 0-ден 9-ға дейінгі сандар. Күш 10 column. Егер біздің нөмірі 130, бұл $1 + 10^2 3 * 10^1 + 0 * 10^0$ мәні ұсынады немесе 1 бағанасында жүздеген, 3-бағанасында ондаған, және 0-бағанасында.

Екпінді екі базасы себептермен негізі 1 тақ болып табылады. Біріншіден, анықталғаннан кейін жоғарыда негізі 1 білдіреді, бұл біздің тек сандар 0, бірақ бұл мағынаны барлық болатын санының мүмкіндігі, айталық 0000, 0000000000, 0, барлығы 0, өйткені формула мәні * қуаты әрқашан. Осы себептен бірлік жүйеде 1 санын 0 орнына қолданамыз. Екіншіден, барлық 1 ($1^0=$, $1^1=1$, $1^2=1$, $1^3=1$). Нәтижесінде көрсетілген сан сандардың сомасы болуы екі талай. Мысалы, бірлік жүйесіндегі 111 ол 3 санын білдіреді, ал 111111111 сандары 9 білдіреді.

5 базасын талқылайық. Бұл негізде біз тек 0-ден 4-ке дейін сандарын және «бағанды» қолданамыз. Оң жақтан сол жаққа дейін 1 бағандық, 5 бағандық, 25 бағандық (2^5), жүз жиырма бес бағанды (5^3), алты жүз жиырма бес бағанды (5^4) және т.б. 1234 саны 5 базада мына түрде көрсетіледі $1*125+2*25+3*5+4*1=194$ ондық бөлшекте.

3.1-кесте Жалпы нөмірлеу жүйесі

Негіз	Атауы	Қолданылған себептер/Таралуы
1	Унитарлық	теориялық информатиканы, машиналармен жұмыс істегенде Тьюринг, сирек пайдаланылады
2	Екілік	цифрлық компьютерлер пайдаланады, сондықтан екілік компьютерлік байланысты пәндер оқытады
8	сегіздік	топтың үш екілік саны сегіздік санын пайдаланылатын компьютерлік байланысты пәндер жасауға аса ыңғайлы оқу үшін екілік жасайды
10	Ондық	Ондық бөлшегін алғашқы адам жүйесі нөмірлейді
16	Он алтылық	он алтылық тобының төрт екілік сан құрып, он алтылық сан болады, пайдаланылатын компьютерлік байланысты пәндер жасау үшін бинарлық астам оқу үшін қолайлы, оңай әрі қарай қарағанда сегіздік (талап аз сан), бірақ неғұрлым күрделі-қосу сан A-F болып табылады.
20	Жиырмасыншы	Колумб кезінен бұрын майя өркениетін пайдаланған; нүктелер мен сызықтар ұсынылған сандар, 0-4 белгіленді, 0-ден 4 нүкте, 5-9 белгіленді 0-ден 4 нүктелердің астын сызумен, 10-14 60 клин жазу белгіленіп, 0-ден 4-нүкте 2 астын оқу, 15-19 белгіленді 0-ден 4 нүктелердің 3 астын оқу; Майя өркениетінен тыс пайдаланылмаған.
60	Алпысыншы	Вавилон өркениетінде пайдаланылған

Алынған кез-келген $a_n a_{n-1} \dots a_1 a_0 b$ базасында, бұл сан мына есеппен шығарылған $\sum_{i=0}^n (a_i * b^i)$ Яғни, біз санды алайық және шеткі оң-ға көбейту бойынша b_0 оны қосу келесі оң жақ санында және көбейту, оны b_1 және оны келесі оң жақ санындағаны қосу және көбейту, оны b_2 , ..., және қосу, оны екінші шеткі сол жақ санында (бағанасында $p - 1$), содан кейін көбейту $BH-1$ және қосу, оның сол жақтағы шеткі санында ($-$ бағанда p) және көбейту, ол млрд есебінде болады.

Қандай базаб илік бағанда қарамастан әрқашан қиыр құқығымен 1? Неге ? Соңғы оң жақ бағана әрқашан қуаты $basecolumn$, онда баған 0 немесе

base0 болады. Математика, кез келген бүтін оң сан (1 немесе одан да көп) дейін артылды, 0-ші қуат әрқашан 1 тең. Өйткені бар шексіз саны жүйелерін нөмірлеу (кез келген құқықтық базасы оң болып табылады), қандай да бірқатар базасын анықтауға біз қабілетті болуымыз керек. Біз, индексінің нөмірінің соңында негізін орналастырамыз. Сонымен 12345 1234 базасында 5, ол біз көрген жоғарыдағы дегеніміз де-19410. Ыңғайлы болу үшін, біз түсіру базасына мүмкіндік беру керекпіз, істің негізі 10 (ондық жүйе), себебі бұл база әдепкі бойынша $1234_5 = 194$.

Негізі 10-нан көп, айталық, база 12 қарастырайық. Нөмірі 246_{12} . Бұл сан мәнінде $* 122 \ 2 + 4 * 12 + 6 * 1 = 342$. Бірақ естеріңізге сала кетейік, кез келген базалық, біз диэлектриктік Ted сандарды пайдаланамыз 0..k - 1 үшін негіздері 12, бұл сандар 0-11. Енді бізде мәселе бар, себебі 10 және 11- біркелкі сандар болып табылады.

Сіздің жазу нөміріңіз 1112 сияқты көрінеді, ал болған түрі 1. Он екі колондық бағанасында 1 орын-өз бағанасында орнына 11 бір мақала бойынша b0 оны келесі оң жақ санында қосу және көбейту, оны b1 және оны қосу келесі оң жақ санында және көбейту, оны b2, ..., байланысқа ресми жарияланған, тапсырудың балама тәсілін сан базасында артық 10 табуға тиіс. Бақытымызға орай біз үшін, әдетте, тек бір негіздеме артық 10, негізі 16 пайдаланады, сондай-ақ белгілі он алтылық сан болады. Осы үшін база туралы шешім қабылданды, бұл сандар ұсыну үшін 10, 11, 12, 13, 14, және 15 қолданылатын болады: A, B, C, D, E және F, тиісінше. Осылайша, мәні 3AB216 құралы 3 163 бағанасында, (немесе 10) 162 бағанасында, (11) 161 бағанында және 2 -+өз жақтық, бағанасында. Бұл 3AB216 білдіреді *

$163 \ 3 + 10 * 162 + 11 * 161 + 2 * 1 = 15,026$.

Есептеудің екілік жүйесі

Бұл неге маңызды? Біздің компьютерлер негізі 10 (он мындық) пайдаланған, бірақ компьютерлер барлық ақпаратты сақтай береді, барлық ақпаратты өңдейді, екілік орнына ондықты пайдаланады. Дегенмен, АТ-маман, екілік кодты деректерді қарап немесе білдіруді алаңдатты, пайдалану ондық емесі жүйелерінің нөмірлеуі келеді. Ең көп таралған пайда болуы ондық емес жүйесін Берингова саны болып табылады, яғни желілік байланыс. IP-мекен кейде екілік жүйеде және IPv6 егжей мекен-жайы - он алтылық жүйесін білдіреді. Әр түрлі компьютерлік қосымшаларда сегіздік пайдаланылады, бірақ бұл өте сирек. Бұл тарауда алдымен екілікті біз қарастырамыз, содан кейін сегіздікті және он алтылық санды. Бинарлы ең маңызды ондық жүйесінен тыс нөмірлеу.

Сандық сипаттағы компьютерлер (яғни, әр бірінде екі әдістері сақталды, қазіргі уақыттағы ток), барлық түріндегі толықтырулар мен ымыраға келуге жол берілмеуі ұсынылуы тиіс. Сондықтан екілік 1 білдіреді және 0 өшіруді білдіреді. Бұл компьютерлік дүкендер алдымен мүмкін болып құрылды: 1s және 0s. Біз атайтындай 1 немесе 0 мәнін бит (екілік сан).

Бір сым үшін әрбір бит орнын ауыстыру деректерді талап етеді, сондықтан беру дерек (айталық 8 бит бір уақытта) 1 бит сымдар емес, 1 бит бір рет жасалуы мүмкін. Сандық сұлбалар БҰ битпен жұмыс істейді, екі бит тіркелімдер және жадтысақтау. Компьютерлік ғалым немесе инженер компьютер. Сіз кейде деректерді немесе бағдарламалық екілік кодты қарауға тиіссіз. Егер сіз мың биттерін қиын болар екенін білсеңіз оған зейін қойыңыз. Сондықтан оның орнына, біздің деректер сынады. Сегіз бит орналастырылуы бірге тиіс блок деп аталады, ол байтпен аяқтайды. Бұл ертедегі компьютерлерде тараған, шамасы сақталуы үшін бір байтте, мүмкін, екі. Байт бола отырып, 8 бит сақтай алады, кез келген әрекеті 8 нөлдердің және бірлік, мысалы, 11111111, 10101010, 10001000, 00000000, 01111110, 01011011.

256 түрлі нөлдердің комбинациялары бар және бірлік 8 бит (1 байт) орналастырылуы мүмкін. Қарапайым есептеулер саны 28 (көбейту екі сегіз рет). Сандық түрлендіру келген екілік санды ондық жүйеде формулалар алдыңғы бет көмегі менжасалуы мүмкін. Дегенмен, бұл туралы ойлауға одан да көп қарапайым тәсілібар.Мысал ретінде байт сақтау 010101012.

Формуласына сәйкес, біз бұл түрлендіру $* 27 + 0 + 1 * 26 + 0 * 25 + 1 * 24 + 0 * 23 + 1 * 22 + 0 * 21 + 1 * 0 = 20 * 128 + 1 * 64 + 0 * 32 + 1 * 16 + 0 * 8 +$

$* 4 + 1 + 0 * 2 + 1 * 1 = 85$. Назар аударыңыз, әрбір сан екілік санынына тең 0 немесе 1 қарағанда айтуға болады, мысалы 5 базасында алдыңғы бөлімде 0, 1, 2, 3 және 4 сандар мүмкіндік берді. Қайта құру кезінде келген екілік санды " он мындық, туралы алаңдатты, көбейту саны қуатты қажеті жоқ. Оның орнына, көрсеткіш 0-бұл $0 * \text{куат} = 0$ және мүмкін кеткендер қосу үшін немесе 1 саны, бұл жағдайда $1 * \text{куаты} = \text{куат}$. Сондықтан бізде жиынтықтау жүргізіледі, тек өкілеттіктерін 2 баған, құрамында 1. мысалы алдыңғы оралу, шын мәнінде алар жазу, 010101012 $64 + 0 + 0 + 16 + 0 + 4 + 0 + 1 = 85$, суретте көрсетілгендей 3.1. Егер біз біле білсек өкілеттігі әр бағанның (2 дәрежелі), онда барлық, қосу осы мәндерін баған байланысты азайтуды, сандар 1s. 3.2-кесте көрсеткендей, көптеген дәрежелі негіздері 2. Сіз бұдан бастап 1, содан кейін екілікті нөмірі үшін әрбір жоғары қуаты (немесе, сіз солға жылжытқан әрбір баған). Мынаған назар аударыңыз, қысқаша ыңғайлы болу үшін

бұл өкілеттіктері 2 тұрғызылған 10 бірлік (10, 20, 30).

$2^{10} = 1024$ немесе 1 К немесе шамамен 1000 (егер сәл дөңгелектеуге біз қарсымыз). $2^{20} = 1$ М немесе шамамен 1 миллион (кестеде көрсетілгендей 3.2 нақты мәні 1048576). $2^{30} = 1$ G немесе шамамен 1 млрд. $2^{40} = 1$ Т немесе шамамен 1 трлн. 2^{50} (кестеде көрсетілгендей) = 1 P PETA үшін, бұл шамамен 1 квадриллион.

2-кестеде 3.2 пайдалану үшін өкілеттік қоюға тырысайық. Ескере отырып, кез келген екілік санының, жай ғана қосу қуаты 2 диаграмма кестеде 3.2 негізінде орналасқан 1S, ондағы сан 20 мәнін білдіреді. Мысалы, 1101 болады егер 23, 22, қамтуға көмек көрсету болса, бірақ 21, 20, немесе $22 + 23 + 20 = 8 + 4 + 1 = 13$. Біз ұстанатын байт ұзындығы (8 бит) саны оңай болу үшін. (8 бит) оңайлық үшін.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
0	1	0	1	0	1	0	1	
	64		16		4		1	= 85

Сурет 3.1 Екілік жүйеден ондыққа ауысу мысалы

3.2-кесте. Әртүрлі деңгейдегі екіліктің мәні

Деңгей 2	Мағынасы
2^0	1
2^1	2
2^2	4
2^3	8
2^4	16
2^5	32
2^6	64
2^7	128
2^8	256
2^9	512
2^{10}	1,024

2^{11}	2,048
2^{12}	4,096
2^{13}	8,192
2^{14}	16,384
2^{15}	32,768
2^{16}	65,536
2^{20}	1,048,576
2^{30}	1,073,741,824
2^{40}	1,099,511,627,776

$$11000110: 27 + 26 + 22 + 21 = 128 + 64 + 4 + 2 = 198$$

$$10101010: 27 + 25 + 23 + 21 = 128 + 32 + 8 + 2 = 170$$

$$01000111: 26 + 22 + 21 + 20 = 64 + 4 + 2 + 1 = 71$$

$$00001110: 23 + 22 + 21 = 8 + 4 + 2 = 14$$

$$00110011: 25 + 24 + 21 + 20 = 32 + 16 + 2 + 1 = 51$$

$$00000000: 0$$

$$11111111: 27 + 26 + 25 + 24 + 23 + 22 + 21 + 20 =$$

$$128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$$

Назар аударыңыз, бұл соңғы жағдайдағы үлкен 8-разрядты сан, оған тең 28 - 1.

Бұл қайта құру туралы ондық жүйенің екілігі. Екі стратегиясы бар, біз де қолдануға болады. Төменде дәстүрлі стратегия, бірақ басқа болуы мүмкін, оңай-стратегия төменде көрсетілген.

Ондық мәнінен мен 2 бөлуді алыңыз. 2 факторын бөліңіз. Жалғастырыңыз, бұл кезге дейін 0 қалып отыр. Жазба қалдықтары бөлу (қалдықты бөлу кезінде 2 немесе 1 немесе 0). Жинау қалдықтарының кері тәртіпте (немесе төменнен жоғары) екілік сандармен мәні болып табылады. Мысалы, тырысайық 89

$$89/2 = 44 \text{ қалдықпен } 1$$

$$44/2 = 22 \text{ қалдықпен } 0$$

$$22/2 = 11 \text{ қалдықпен } 0$$

$$11/2 = 5 \text{ қалдықпен } 1$$

$$5/2 = 2 \text{ қалдықпен } 1$$

$$2/2 = 1 \text{ қалдықпен } 0$$

$$1/2 = 0 \text{ қалдықпен } 1$$

(біз 0 қол жеткіздік). Екілік эквиваленті 89 01011001 (онда осталь - Ders кері тәртіпте, қосымша сәл алда жасауға, оның 8-биттік саны). Біз бұл айырбастау арқылы бұл санды қайтадан ондық жүйесін растаймыз:
 $01011001 = 64 + 16 + 8 + 1 = 89$, 251-бабымен тағы бір тырысайық:

$$\begin{aligned} 251/2 &= 125 \text{ қалдықпен } 1 \\ 125/2 &= 62 \text{ қалдықпен } 1 \\ 62/2 &= 31 \text{ қалдықпен } 0 \\ 31/2 &= 15 \text{ қалдықпен } 1 \\ 15/2 &= 7 \text{ қалдықпен } 1 \\ 7/2 &= 3 \text{ қалдықпен } 1 \\ 3/2 &= 1 \text{ қалдықпен } 1 \\ 1/2 &= 0 \text{ қалдықпен } 1 \end{aligned}$$

осылайша, 251 11111011. Тексеру үшін біз артқа жібереміз, $128 = 11111011 + 64 + 32 + 16 + 8 + 2 + 1 = 251$. Тәсіл бөлінуі оңай, бірақ интуитивті болу міндетті емес, өйткені сіз өзгерту тәртібі қалдықтарын екілік санын құрайды. Басқа тәсілі түрлендіру үшін келген ондық жүйенің екілік пайдалану есептелген болып табылады. Ескере отырып, ондық саны анықтау, оған ең жоғарысын қалдырып, дәрежесі 2 шегеріліп тасталуы мүмкін, оң сан немесе 0. Мұны әзірге сіздің ондаық саны 0-ге тең. Әр дәрежелі 2 саны шегеріледі.

Тиісті бағанда 1 жазу. Мысал ретінде сол құндылығымызды қарастырайық, оны біз тек қайта құрдық, 251. 3.2 кестеге қайта оралсақ, ең үлкен дәрежесі 2, біз 251 128 шегереміз. " 128-і 251 есептеу, бізде 123. Ең үлкен күші 2 көрініп тұрғандай, шегеруге 123 64, нәтижесінде 59. Ең үлкен дәрежесі 2, біз шегеруге 59 32 27. Қалдырып ең үлкен дәрежесі 2, біз шегеруге 27 16 11. Шығу ең үлкен дәрежесі 2, біз 11 шегеруге болады: 8 қалдырып, 3 аламыз. Ең үлкен қуаты 2 біз бағасын 3 болып табылады 2 кеткен 1. Ең үлкен қуаты 2 біз шегеруге 11-ге тең 0 қалдырдық. Біз аяқтадық. Біздің түрлендіру процесі бұл: екілік нөмірлеу жүйесі 63

$$\begin{aligned} 251 - 128 &= 123 \\ 123 - 64 &= 59 \\ 59 - 32 &= 27 \\ 27 - 16 &= 11 \\ 11 - 8 &= 3 \\ 3 - 2 &= 1 \\ 1 - 1 &= 0 \end{aligned}$$

Осылайша, $251 = 128 + 64 + 32 + 16 + 8 + 2 + 1$. Осылайша, біздің екілік баламасы бар 1s қақтығусуда

ВМН білдіретін 128, 64, 32, 16, 8, 2 және 1 және 0s басқа жерде (тек басқа баған ұсынады 4). Бұл бізге 11111011. Біз тексеру жұмысымызды 3.2-суретте көрсетілген, ол түрлендіреді 11111011 кері ондық жүйесі, 251.

Тағы бір мысал. Түрлендіру екілік жүйеде 140. Ең үлкен күші 2, біз шегеруге 140 128 қалдырып 12. Ең үлкен қуаты 2, біз шегеруге 12 8 4 қалдырып. Ең үлкен қуаты 2, біз шегеруге 4 4 қалдырып 0.

$$140 - 128 = 12$$

$$12 - 8 = 4$$

$$4 - 4 = 0$$

$140 = 128 + 8 + 4$. Осылайша, біз 1s-бағаналарда, олар 128, 8 және 4-0s басқа жерде.

Бұл мән 10001100 көрсетті, сондықтан $140 = 10001100$.

Әлі күнге дейін, біз екілік санының 8 битін шектедік. Естеріңізге сала кетейік, 8 бит, біз 256 әр түрлі комбинациялары 1s және 0s. Өйткені 00000000 0-ге тең болса, онда 8 бит (1 байт) ауқымы-0-ден 255. Егер біз сақтауға көп? Содан кейін бізге көп бит. Мысалы, 2 байт (16 бит), біз сақтауға, кез келген санды 0-ден 65535. Естеріңізге сала кетейік, $216 = 65536$. Жалпы жағдайда, егер біз N бит, біз сақтауға, кез келген санды 0-ден $2^n - 1$. Біз байт ретінде ыңғайлы жад көлемі (орнына 1 немесе бірнеше бит) жиі пайдаландық, шектеп өзіне 1 байт болып табылады тым шек кою. Қазіргі заманғы компьютерлер әдетте 32 битті қамтамасыз етті (4 байта) немесе 64 битті (8 байт) типтік мөлшерін сақтау үшін. Біз типтік түрі жад мөлшері ретінде компьютер сөздерін алдық .

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
1	1	1	1	1	0	1	1	
128	64	32	16	8		2	1	= 251

Сурет 3.2 Жауапты тексеру үшін артқа өзгертеміз

Қаншалықты 32 бит зор? Бұл мәндер аралығы:

$$00000000000000000000000000000000 = 0$$

$$11111111111111111111111111111111 = 4294967296 \text{ (шамамен 4,3 миллиард)}$$

Көптеген қосымшалар, бізге жеткілікті 32 бит орын сақтау үшін 64 бит, ең көп саны - 18 квадриллиондарды (18 жүреді 15 нөлдермен!) сақтайды. Бұл үлкен сандар 32 - немесе 64-биттік болып табылады, сөз өлшемі ысырапқорлық болатынын еске алу керек. Бұл қосымшалар. Дегенмен, жады көлемі кемінде 4 миллиард байт және дискілік сақтау жартысынан кем емес триллион байт пайдалана отырады, 32 немесе 64 бит сақтау үшін алаңдаушылық тиіс нөмірін туғызады. Бірқатар сандарда сақталатын 32-биттік екілік мәндерді қарастырайық. Бұл сандар 1 және 0 тең, нәтижесінде тізім өте қиын дерлік ретінде қарап, оптикалық иллюзия. Ыңғайлы болу үшін, біз екілікті сандарды әр түрлі көріністерде аударамыз, әдетте (негізі 8) немесе он алтылық үлгіде болып табылады (негізі 16). Осы материалдың беттерін елестетіңізші!

```
11101100100010110010101100000111
00101100010100101010110110010001
10001011001001001011011100100111
10001000101001111010001110101011
00001001010100000111100010101100
```

Бұл, әрине 8 және 16 есе көп 2. Бұл аудару 2-ден 8-ге дейін негіз болады, немесе негізі 16, қарапайым (салыстырғанда аудармасымен негізі 2-ден 10 немесе негіз 10 табанына 2). 8 алу 3 екілік разрядтар негіздері үшін және оларды топтау болады, содан кейін қайта, бұл 3 бита келген екілік болып табылады. Ол жұмыс істейді, өйткені $2^3 = 8$. Ескереміз, сегіз санын пайдаланады (0-ден 7-ге), 3 битті конверсиялау келген екілік болып табылады, сол сияқты 3 битті түрлендіру бірі-екілік ондық санау жүйесі. 3.3 кестеде көрсетілгендей, 3 бит арасындағы екі негіз түрлендірді.

Кесте 3.3 Екілік кодтан сегіздікке ауысу кестесі

Екілік	Сегіздік
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Бірінші 32-биттік нөмірі алдыңғы беттерін пайдалана отырып, түрлендіру болып табылады. 11101100100010110010101100000111

Біріншіден, 32-разрядты саны 3-биттік сегменттерінен бастап бөлу, оң аяғына 101 100 11 100 010 110 010 101 100 000 111

Екіншіден, өйткені сегмент бұл соңғы сол 3 бит, біз айта кетейік жетекші 0s үшін оны 3 бит 011 100 100 101 010 110 010 101 100 000 111

Үшіншіден, пайдалана отырып, жоғарыда келтірілген кестені, біз әр топқа 3 битті түрлендіру, оның баламасы сегіздік маңызы бар

3 5 4 4 2 6 2 5 4 0 7

Және ақыр соңында, біз жекелеген сегіздік сандарды бір мәнге біріктіреміз: 35442625407. Осылайша, 11101100100010110010101100000111₂ = 35442625407₈.

Қалған төрт саны былайша түрлендіріледі:

00 101 100 010 100 010 110 101 110 010 001 = 05424526621₈
 10 001 011 001 001 011 011 001 100 100 111 = 21311133447₈
 10 001 000 101 001 010 001 111 110 101 011 = 21051721653₈
 00 001 001 010 100 111 100 000 010 101 100 = 01124074254₈

Бүкіл сандар тізімі

35442625407

05424526621
 21311133447
 21051721653
 01124074254

Көп сандар біздің сегіздік сандарды аударуға күресуде. Біз мөлшерін одан қайта құру жолымен тізімін екілік санынан он алтылық жүйесін азайтамыз. "Он алтылық форматында біз 4-топ-разрядтық екілік санының түрлендіруі, оларды он алтылық ондық белгіге баламалы. 3.4-кестеде келтірілген екілік, он алтылық және ондық баламалары үшін жекелеген он алтылық сан (алғашқы 0-15 сандық ондық жүйесі). Назар аударыңыз, бұл 10-15 үшін, біз A-F болады, сондықтан біз осы мәндер ретінде сақтауға жекелеген сандарды бұрын талқыладық.

Кесте 3.4 Екілік жүйеден он алтыншы жүйеге ауысу

Екілік	Он алтылық	Ондық
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Сол нөмірді бұрынғыдай пайдалануға болады, 111011001000101100101011000001112 және он алтылық санды конверсиялаймыз. Бұл тәсілді қайта құру кезінде біз басқа келген екілік тобының 4 биті болып табылатынын жасадық. 3.4 кестеде көру үшін он алтылық сандар белгіленген.

1110 1100 1000 1011 0010 1011 0000 0111

Е С 8 В 2 В 0 7

Жоғарыда көрсетілгендей,

$11101100100010110010101100000111_2 = EC8B2B07_{16}$.

Басқа сандар біздің тізімде төмендегідей қайта құрылатын болады:

$0010\ 1100\ 0101\ 0010\ 1010\ 1101\ 1001\ 0001 = 2C52AD91_{16}$

$1000\ 1011\ 0010\ 0100\ 1011\ 0111\ 0010\ 0111 = 8B24B727_{16}$

$1000\ 1000\ 1010\ 0111\ 1010\ 0011\ 1010\ 1011 = 88A7A3AB_{16}$

$0000\ 1001\ 0101\ 0000\ 0111\ 1000\ 1010\ 1100 = 095078AC_{16}$

Кері екілік мәні үшін түрлендіру болып табылады, жай деген әрбір сегіздік санына ауыстырылсын, оның үш таңбалы екілік баламасы болып келеді. Дегенмен, сіз барлық 3 сандары екілік жүйеге көз жеткізіңіз. Мысалы, $7238 = 111\ 010\ 0112$, $111\ 10\ 112$. Мәні $123456708 = 001\ 010\ 011\ 100\ 101\ 110\ 111\ 000 = 0010100111001011101110002$, бұл 32 битаға тиіс сақталған, біз айта кетейік жетекші салықтарын ұзартуға, оның 32 биті бізге не береді:

$00000000001010011100101110111000_2$

Он алтылық мәні үшін екілікті түрлендіру, әрбір он алтылық саны 4 эквивалентті-разрядтық екілік маңызы бар. Мысалы $2468BDF016 = 0010\ 0100\ 0110\ 1000\ 1011\ 1101\ 1111\ 0000_2 = 0010010001101000101111\ 0111110000_2$. Артықшылығы сегіздік немесе он алтылық санды пайдалануға болады. Үстінен екілік болып табылады, яғни бұл сандар оңай.

ТЕРІС САНДАР МЕН БӨЛШЕКТЕР

Санаудың екілік жүйесінде, біз кез келген бүтін он санды көрдік (бүтін сан), қайта құрылуы мүмкін, бірі ондық санау жүйесі екілік болады. Барлығы бізге керек, бұл жеткілікті саны, бит сақтау үшін екілік сандар. Біз теріс сандарды ондық нүктесінде қалай сақтаймыз? Біз де әр түрлі ұсынысты пайдалану керекпіз.

Теріс сандар үшін, компьютерлер ұсыну екілік толықтыру деп аталатын. Ал екі-х комплемент біркелкі нөмірі әдеттегі екілік сан оң мәні болады. Егер сан теріс болса, біз орындауды түрлендіреміз. Түрлендіру талап етеді, сіз қабылдауға оң екілік сан барлық битке бұрасыз (0s айналады

1s, 1s болуға 0s), 1қосыңыз. Бинарлық операциялар қосуды қарастырады, бірақ біз оны қолданамыз, мұнда барлық мысалдар қарапайым боып келеді.

Елестетіп көріңізші, бізде саны 61, біз сақтау үшін, оның 8-биттік екі комплемент маңызы бар қоршаған орта бар. Өйткені ол оң, біз қалыпты ондық "екілікті түрлендіру: $61 = 32 + 16 + 8 + 4 + 1 = 00111101$, сондықтан 61 00111101.

Қалай -61? Өйткені $61 = 00111101$ алу үшін -61, барлық битті аудару қосу 1. Соқалар беруге $00111101\ 11000010$ бізге (оңай әрбір 0, 1 және әрбір 1 A 0). Енді қосыңыз $1-11000010$, және сіз 11000011 . Сондықтан $-61 = 11000011$. Назар аударыңыз, жетекші бит 1-ге тең. Қосымша бойынша теріс сан әрқашан басталуы және оң саны 1 әрқашан басталуы 0.

Тағы бір рет тырысып көрейік, 15. Бұл $8 + 4 + 2 + 1 = 00001111$. Сондықтан -15 барлық соқалар алу үшін 11110000 өзгерту қажет, содан кейін қосу 1 алу үшін 11110001 . Тағы да, назар аударыңыз оң хабары басталады 0, теріс-нұсқа 1.

Қайта құру үшін екілік ондықта санау жүйесі, саны оң болып табылады, әдетте ол жай түрлендіру болып келеді. Мысалы, 00111101 оң, сондықтан саны болады $32 + 16 + 8 + 4 + 1 = 61$, және 00001111 болымды, сондықтан саны болады $8 + 4 + 2 + 1 = 15$.

Қалай 11000011 ? Бұл-теріс сан. Біз сол техниканы түрлендіреміз, біз түрлендіру үшін келген ондық жүйенің екілігін пайдаланамыз: өзгерте барлық бит қосу 1. Сонымен 11000011 айналады $00111100 + 1 = 00111101$, ондық саны 61, сондықтан 11000011 тиіс -61. Және 11110001 теріс, сондықтан біз барлық бит қосу аударамыз 1 бізге $00001110 + 1 = 00001111$, ондық сан 15, сондықтан 11110001 тиіс -15.

Енді түрлендіруге тырысайық -48 в екілік форматы. Біз қайтадан бастаймыз $+48 = 32 + 16 = 00110000$. Өйткені біз -48, барлық бит өзгертіп қосыңыз. Барлық соқаларды ақтарыңыз, біз көшеміз 00110000 дейін 11001111 . Енді біз 1 қосамыз. Бұл $11001111 + 1$? Ал, бұл талап арифметиканы шешу үшін сәл екілік керек. Бұл қиын емес; сіз жай ғана кейбір автокөлік - RIES орындау керексіз, бұл сізге 11010000 . Сонымен -48 болып табылады 11010000 . Өйткені екілік арифметика бинарлық операцияларға қаралады, қазіргі уақытта, біз баламалы тәсілі теріс сандарды екілік жүйеде түрлендіреміз. Біз қайтадан теріс санның ондық жүйесін бастаймыз. Түрлендіру оң нұсқасын екілік ретінде біз алдыңғы жақындаған күні жасадық. Біз он екілік сан оның теріс баламасы түріндегі толықтыруға дейін түрлендірдік. Енді сіз осынау қарапайым стратегияны қолданасыз:

1. Оң жақ санында жұмыс істей отырып іске қосыңыз, оңнан солға қарай, көшіруге, әрбір 0, егер сізге бірінші 1 бит жетсе
2. Көшіру және қою бірінші 1 бит
3. Жұмысты жалғастыру, оңнан солға қарай, қалған барлық биттерге аудару, мысалы, 00110000 болып төмендегідей қайта құрылады:
 1. Оң жағынан бастап, біз қалған құжаттар төменге барлық 0s
... 0000
 2. Біз біріншіге қол жеткізгенде 1, оны көшіріңіз
... 1 0000
 3. Қалған барлық флип биттер, бұл жағдайда 001 айналады 110
110 1 0000

Осылайша, түрлендіру 00110000 оңға дейін теріс бізге 11010000. Егер саны бар 1-қиыр құқық орнында? 11000011 (-61) Нөмірімен бастаймыз. 1-қадам емес пайдаланылады, себебі 0s жоқ; сондықтан, оның орнына, біз шеткі оң жақ 1 жазамыз, содан кейін қалған соқалар бере отырып аударыңыз, бізге 00111101 (+61).

Бөлшек санының қарапайым және неғұрлым күрделі түсіну үшін. Олар оңай

түсіну болып та $\sum_{i=0}^n (a_i * b^i)$ кені жалғасы мына формула бойынша ұсынылған бұрын, қоспағанда сонымен қатар, біз таратамыз I қосу теріс сандар үшін ғана пайда болады, оң жағында ондық. Сіз еске негізгі алгебраны, б-мен = 1 / би. Осылайша, В-3 шын мәнінде 1 / В3 санаттарына қолдануға арналған. "Екілік түріндегі, Б әрқашан 2, сондықтан біз кейбір теріс дәрежесі зерттейміз 2. 3.5-кесте бірнеше теріс дәрежелі 2 көрсетеді.

Кесте 3.5 Теріс (бөлшектер) өкілеттіктер 2

Экспоненталық пішін	Бөлшек түріндегі мағына	Нақты санның сапасындағы мағына
$2^{-1} = 1/2^1$	1/2	0.5
$2^{-2} = 1/2^2$	1/4	0.25
$2^{-3} = 1/2^3$	1/8	0.125
$2^{-4} = 1/2^4$	1/16	0.0625
$2^{-5} = 1/2^5$	1/32	0.03125
$2^{-6} = 1/2^6$	1/64	0.015625
$2^{-7} = 1/2^7$	1/128	0.0078125

Мысалды қарастырайық. Бұл екілік сандармен болып табылады ма 110111,01?, біздің формуланы жоғарыдан пайдалана отырып, $110111,01 = 1 * 25 + 1 * 24 + 0 * 23 + 1 * 22 + 1 * 21 + 1 * 20 + 0 * 2^{-1} + 1 * 2^{-2} = 32 + 16 + 4 + 2 + 1 + \frac{1}{4} = 55,25$.

Тағы бір мысал $101,101 = 4 + 1 + \frac{1}{2} + \frac{1}{8} = 5,625$. Сіз мынадай мысалдар әзірлейсіз бе?

1111.1001
0,1111
10101.101

Түрлендіру бірі-екілік санау жүйесі ондықта өте қиын болып келеді. Дегенмен, оның мәселесі бар. Біз, мұны компьютерге сақтауға 55.25 мәні сияқты болғанын қалаймыз, ол ауыстырылуы екілік болуы тиіс, осылайша біз 110111.01. Дегенмен, компьютер тек 1 және 0 сақтайды, біз қалай ондық нүктесін көрсете аламыз ба? Ондық нүктесінбіз сақтауға тырысамыз, бірақ жазба түрінде ондық тиіс.

Ондық үшін белгілеуге, біз неғұрлым күрделі стратегиясына негізделген ғылыми түрде ескереміз. Ғылыми аннотациясында нақты түрінде құны $55,25 \cdot 0,5525 * 102$. Яғни, ондық нүкте алдында бірінші нөлдік сандарды жылжытады. "Ондықығыстыру, алайда, біз көбейту маңызы дәрежесі 10-ға тең. Осылайша, $55,25$ болады $0,5525 * 102$. Енді бізде екі санын 0.5525 және 2 сақтау үшін. Біз ондық нүктесі ретінде өткізіп түсінеді, яғни үтірдің алдында әрқашан бірінші нөлдік сандар болады. Біз сондай-ақ 10 төмен, өйткені ол, бұл базада 10 көздейді.

Екілік түріндегі ең негізгі нәрсе. 110111.01 бастайық, ондық нүктесін жылжытуға алдында нөлдік сандар (бірінші 1) көбейту арқылы дәрежесі 2. Содан кейін біз ондық нүктесін түсіреміз және 2 сақтау нөмірі түріндегі екі бөліктен. Пайдалана отырып 110111.01 біздің мысалда, алдымен жылжытуға ондық нүктесін бере отырып, бізге $0,11011101 * 2^6$. Біз 2^6 , біз ондық үтірден 6 позицияға көштік. Өйткені біздің нөміріміз аз болды (110111.01 дейін .11011101) үлкен көлемді біз көбейттік. Қазіргі уақытта, бұл түрінде жай ғана екі саннансақтауға болады, 11011101 және 110 (бұл көрсеткіш 6). Мәні 2 пайдаланылады көбейту базалық болып табылады. Өйткені негіз бойынша 2 түрі бар, біз оны алып тастау үшін біздің сандар шын мәнінде, қосу үшінші қатар, біздің қойма, таңбалық разряд. Біз 0 пайдалану үшін, оң және 1 -теріс. Осылайша, біздің санымыз 0 белгілі битке тең, 110 көрсеткіш үшін және 11011101 нақты саны. Бұл түрінің бір бөлігін мантиссалар деп атаймыз.

Біз үшінші бөлігінде сол жад ретінде белгілі битті, бит экспонатты, бит

мантиссасын ұйымдастырамыз. Таңбалық бит әрқашан бір бит болып табылады. Біз бита мөлшерін, экспоненттер және мантистарды таңдаймыз. Біз мөлшерлі өкілдіктер әсерін дәлдігіне таңдаймыз. 16-биттік қалқымалы нүктелі дық ұсыну болып көрінеді, оның облысы С 55.25 Біздің алғашқы ондық санын түрлендіру, оны екілік ретінде 110111.01. Енді біз ондық нүктесіне көшеміз $11011101 * 2^6$. Біз экспонентті " екілік пайдалануды 5 битке айналдырамыз, 00110. Және, ақыр соңында, біз мантиссаларды қайта құрамыз, 10 бит бізге 1101110100 болады. Назар аударыңыз, біз 0s деп - мантиссаларды соңында нөлдер дедік.

Бөлшек нөмірі, қосу нөлдердің соңында мәні өзгертпейтінін білдіреді. Біз жетекші нөлдер көрсеткішіне қосылдық, себебі көрсеткіш фракциясы болып келеді. Бұл қосу - ИНГ 0s дейін 12.34, қосу жетекші нөлдер сол жағында барлық бөліктері тұйықтаушы нөлдер оң жағында бөлшекті бір бөлігін бере отырамыз, мысалы 0012.340000. Енді өзгермелі нүктесінің мәні үшін 55.25 0 (таңбалық бит), 00110 (бит экспонат), 1101110100 (мантисса). Бұл 16-битті маңызы бар түрінде сақталады 0001101101110100.

Басқа ондық саны екілік формат арқылы 16-биттік құбылмалы үтір ұсынылады, түрлендіреді. Біздің тырысу бойынша -14.75. Біріншіден, біз назарға осы белгіні аламыз. Біз 14.75 бұл-бинарлық кодын мынадай түрде айырбастауға болады: $14,75 = 8 + 4 + 2 + 0,5 + 0,25$ немесе $8 + 4 + 2 + 1/2 + 1/4$. Бұл екілік сан 1110,11. Енді, біз ондық нүктесін алдында бірінші 1 бізге ауыстырылады. $111011 * 2^4$ осылайша, мантисса 111011 біздің көрсеткіш 4 немесе 100. Мантиссадан 10 бит жалғастырамыз, 1110110000 беріледі, біздің көрсеткіш 5-соқалар бізге беретін, 00100. Ақырында, біз теріс сандармен бастадық, сондықтан белгісі 1-ге тең бит. Осылайша, $-14,75 = 1\ 00100\ 1110110000 = 1001001110110000$.

Біз тағы бір түрлендіру үшін келген ондық санау жүйесі " екілікті жасаймыз. 0,1875 " екілікке айырбастауға болады. Бұл өте қиын көрінеді, өйткені барлық бөлігі біздің нөміріміз. 0,1875 тұрады $1/8 + 1/16$ немесе екілік санының 0,0011. Естеріңізге сала кетейік, біз ондық нүктесін ескерту біріншіге 1 ауыстырамыз. Бұл жағдайда біз ондық нүктесін оң жақта емес, сол жақтағына ауыстырамыз. Осылайша, 0,0011 көбейіп, біз көрген алдыңғы екі мысалдарда аз емес. Сондықтан да бізге 2 дәрежелі теріс көбейту, оң күші 2. $0,0011 = 0,11 * 2^{-2}$. Біздің көрсеткішіміздің саны теріс болып табылады. Біз оң және екі теріс сандар түрінде толықтыруды сақтауға болады. 5 бит, $-2 = 11110$. Осылайша, біздің көрсеткішіміз 11110 болып табылады. Біздің бит белгісі 0-ге тең. Және, ақыр соңында, біздің мантиссаға дейін ұзартылып, 10 бит құрайды 1100000000. Осылайша, $0,1875 = 0\ 11110\ 1100000000$ немесе 0111101100000000.

Енді түрлендіру біздің 16-биттік қалқымалы нүктелі ұсыну кері ондық жүйесі болады. Біз нөмірін 1001111100010110 пайдаланатын боламыз. Екеуіне де бұл нөмір өзінің үш бөлігі 1 00111 1100010110 болып табылады. Бастауыш бізге 1 дейді, бұл-теріс сан. Келесі тобы сан көрсеткіші болып табылады. 00111 екілік код 7, сондықтан біздің көрсеткішіміз 7 білдіреді, біз оңға ондық 7 позицияға ығысамыз. Және, ақыр соңында, соңғы тобы мантисса сандар болып табылады. Ондық нүкте бұл сандарды алдын ала ескертеді. Сонымен 1100010110 шын мәнінде $0,1100010110 * 200111$. Біз 200111 көбейтеміз, 1100010.110 аламыз. Енді біз бұл санды бірі-екілік санау жүйесінде ондықты қайта құрамыз. Бұл сан $64 + 32 + 2 + \frac{1}{2} + \frac{1}{4} = 98,75$, ал белгісі 1бит, бұл сан -98,75 болады. Міне тағы бір мысал, 0111011010000000, 0 11101 1010000000. Бұл оң саны болып табылады. 11101 көрсеткіші болып табылады, теріс сандармен толықтыру түрінде екіге айналдырамыз. Біз оның оң нұсқасын толықтырумен екі, 00011, немесе 3. Осылайша, көрсеткіш шын мәнінде -3 білдіреді, біз ондық үш позицияны сол жаққа ығысамыз (оң жақта). Мантисса +0,10100000000, немесе ыңғайлы болу үшін 0,101. Осылайша, біздің нөміріміз $0,101 * -3 = 0,000101$. Біздің екілік сандардың ондық түрлендіру алгоритмі, $1 = 0,000101 * 2^{-4} + 1 * 2^{-6} = 1/24 + 1/26 = 1/16 + 1/64 = 3/64 = 0,046875$.

16-биттік ұсыну шын мәнінде компьютерлерде пайдаланылмайды. Бар екі танымал форматтарда ұсынбаған немесе қазіргі заманғы компьютерлер пайдаланылатын. Оларды жиі дара дәлдікпен, екілік дәлдікпен деп атайды. "Дара дәлдік белгісі, 32 бит экспоненттер және мантистар үшін пайдаланылады. Қос дәлдігі 64 бит пайдаланылады.

Рәміздердің сипаттамалары

Компьютерде мәтінді сақтау үшін басқа нөмірлерді беруді қалаймыз. Біз мәтінді сақтауға, оның жекелеген нышандарын сындырамыз. Әрбір белгі жеке жад көлемі ретінде сақталады (мысалы, белгі 1 байт). Осылайша, осы тараудың мәтіні түрлендіріледі, әрбір оның ұсынған таңбаларын талап етеді, содан кейін әрбір ұсыну рәміздері байтты жадта сақтайды. Кейіпкер ұсынымдарды қамтуы тиіс ғана емес, әліппенің 26 әріптері, жоғарғы немесе төменгі тіркелімде болады (барлығы 52 әріп), 10сан (мысалы, пошталық индекс немесе телефон нөмірі арқылы біз сандарға сілтейміз), #, \$, &, *, +, тыныс белгілері сияқты бос кеңістік (бос орын) және жолдың соңына (қайтару ақырлы немесе пернесін іске қосу). Бар нысанды ASCII белгілерін ұсынамыз (Американдық стандартты код үшін ақпаратпен алмасу) және Unicode. Үшінші түрі EBCDIC, кеңінен

1960-шы және 1970-шы жылдары қолданылды. Біз қысқаша бұл үш нысандарды қарастырайық:

Біреу ойлап шығарған техногендік әрбір нысандарын ұсыну болып табылады. Әрбір жағдайда бұл белгіні пайдалана отырып сақталатын болады, нақты екілік разрядтарға қарамастан, бұл бит үшін кодтаудың үлгілері маңызды болып табылады. Екі, үш көрініс 3.3 және 3.4 суретте көрсетілді.

Сурет 3.3 ASCII кестесі.

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

Оядық	Сипаттама	Оядық	Сипаттама	Оядық	Сипаттама	Оядық	Сипаттама
000	NUL	043	CU2	127	"	210	K
001	SOH	045	ENQ	129	a	211	L
002	STX	046	ACK	130	b	212	M
003	ETX	047	BEL	131	c	213	N
004	PF	050	SYN	132	d	214	O
005	HT	052	PN	133	e	215	P
006	LC	053	RS	134	f	216	Q
007	DEL	054	UC	135	g	217	R
008	GE	055	EOT	136	h	224	\
009	RLF	059	CUB	137	i	226	S
010	SMM	060	DC4	145	j	227	T
011	VT	061	NAK	146	k	228	U
012	FF	063	SUB	147	l	229	V
013	CR	064	BLANK	148	m	230	W
014	SO	074	€	149	n	231	X
015	SI	075	.	150	o	232	Y
016	DLE	076	<	151	p	233	Z
017	DC1	077	(	152	q	240	0
018	DC2	078	+	153	r	241	1
019	TM	079		161	~	242	2
020	RES	090	!	162	s	243	3
021	NL	091	\$	163	t	244	4
022	BS	092	*	164	u	245	5
023	IL	093	)	165	v	246	6
024	CAN	094	;	166	w	247	7
025	EM	095	~	167	x	248	8
026	CC	096	-	168	y	249	9
027	CU1	097	/	169	z		
028	IFS	106	!	192	{		
029	IGS	107	,	193	A		
030	IRS	108	%	194	B		
031	IUS	109	_	195	C		
032	DS	110	>	196	D		
033	SOS	111	?	197	E		
034	FS	121	'	198	F		
036	BYP	122	:	199	G		
037	LF	123	#	200	H		
038	ETB	124	@	201	I		
039	ESC	125	*	208	]		
042	SM	126	=	209	J		

3.4 Сурет. EBCDIC кестесін көрсетеді. (Отwww.LookupTables.com, http://www.lookuptables.com/ebcdic_scan_codes.php. рұқсаты бойынша ұсыну).

Кейбір рәміздер басқарушылар немесе басып шығарылмаған рәміздер болып табылады, сондықтан кестеде көрсеткішіне назар аударыңыз.

EBCDIC ғана үлкен ЭВМ, IBM пайдаланылады және ішінара тоқтатылады, өйткені бұл өте қисынды ұсыну. Егер сіз қарасаңыз ASCII арқылы ұсынуды сіз көресіз, бұл кез келген екі әріптен ASCII тұрады, сол тәртіппен түбегейлі әріптер әліпбиіге енгізілді; мысалы, "a" тікелей алдында 'B', ол тікелей алдында 'c', және осылайша әрі қарай жалғаса береді. Осылайша бас әріптер алдында кіші әріптермен пайдаланады. EBCDIC дегенмен, әріптер ретімен ('a' алдында " 'B' - дан 'c'), олар міндетті түрде бірден бірімен жүреді, мысалы 'J' ұстануы бірден "мен" ге ауысады. ASCII тәсілі жұмыс істейді, оның орнына белгіні қойған дұрыс, біз екілік реттілігімен жұмыс істейміз. Мысалы, 'a' ондық маңызы бар 97 түрде сақталады немесе он алтылық мәні 61 сан болады. Ондықты пайдалана отырып "екілік немесе он алтылық", бұл таба 'a' 01100001. Салыстырыңыз, бұл 'A' 65 ондық (он алтылық 41), ол 01000001 болып табылады. Бір қызығы, "a" және "a" бірдей дерлік, жалғыз айырмашылық үшінші битпен сол жақта тұрады. Жоғарғы регистр микроскопиялық әрбір әріп үшінші бит 0 , ал төменгі нұсқасы әрбір жағдайда үшінші әріп 1бит болады.

ASCII 7 бит пайдаланады, бізге 128 түрлі қиыстыру берілмейді. 52 әріп 10 сан, сондай-ақ әр түрлі тыныс белгілері бар. Оларға басқарушы ретінде мыналар жатады: табуляция пернесі мен қайтару (немесе жолдан "соңы" деген). Онда 1 байт сақталады, сондықтан соңғы 0 бит 0 болады. Қосу ретінде әкелетін бит (сол жақ шетінде). EBCDIC 8 битті пайдалануды ұсынады және құрамында 256 әр түрлі қиыстырулар бар. Мысалды қарастырайық: біз "Hello"жолын сақтаймыз.Бұл қалай жасалады? "ASCII таңбалардың әр ұсынылған басқа да коды бар. "H" 01001000 болып табылады, "e" 01100101 және 01101111 екі рет кездеседі. Осылайша, "Hello" түріндегі реттілігі бес 8-биттік тізбектер сақталады немесе бес байт: 01001000 01100101 01101101 01101101 01101111 болады. EBCDIC көп жадты (бес байт) сақтау үшін "Hello" көріністері ерекшеленеді. Біз EBCDIC орнына: 11001000 10000101 10010111 1001011110010110 қойамыз. Бірде-ASCII, бірде-EBCDIC қандай да бір табыс ету үшін әріптерден тыс ағылшын әліпбисі қолданылады. Халықаралық сипаты бойынша бүгін есептеулер қажеттілігі бар басқа тілдер рәміздерін ұсынады. Мысалы, араб әріптері, кириллица немесе орыс, қытай және жапон болды. Дегенмен, көптеген компьютерлік пайдаланушылар ағылшынды қолданады.

Сондықтан кеңейту туралы ASCII 7 битті және 16 битті шешімді қабылдайды, бізге 65,356 түрлі комбинациялары беріледі және сақтау мүмкін-

дігі 65000 рәміздер. Бұл UNICODE. UNICODE-та бірінші 128 белгілер және ASCII бірдей. Дегенмен, қалған рәміздерді сақтау үшін басқа тілдерде табылған, сонымен қатар басқа да белгілерімен пайдаланылады. Бұл басқа зодиак элементтер таңбаларды қамтиды, таңбалар үшін шахмат фигурасының белгісі, бейбітшілік белгісі, инь / янь смайлик / frownpu тұлғаның валюта рәміздері, математикалық белгілер мен бағыттама-лар. Алдыңғы "Hello" Unicode мысал үшін көрінді, өйткені бірінші 128 Unicode символдарың 128 белгілері ASCII. Дегенмен, UNICODE әрбір 16 бит емес, 7 символ сақталады. Мәселен, "Hello" сияқты көрінеді:

```
0000000001001000 "H"
0000000001100101 "e"
0000000001101101 "l"
0000000001101101 "l"
0000000001101111 "o"
```

Осылайша, "Hello" Unicode-ты екі есе көп сақтауға талап қойды, бұл кез келген ASCII немесе EBCDIC. Осы тараудың соңында біз басқа қолдану жетекші битті ASCII (сол жақтағы шеткі бит, біз 0 қойдық) табамыз.

Екілік операциялар

Басқа сандарды түрлендіруі, компьютерді өңдеуі тиіс. Әдетте, біз ойлаймыз компьютерлер операцияларды орындайтын косу тәрізді азайту, көбейту, бөлу және салыстыру (екі мәндерін анықтау үшін салыстыру, егер олар тең немесе аз болса, ол басқа). Дегенмен, тек екілік маңызы бар компьютер жұмыс істейді және бинарлық операцияны орындайды (бұл, булелді операциялар). Түрлендірудің өзінің қарама-қарсы маңызы бар (1 тең 0, 0 айналуға 1)

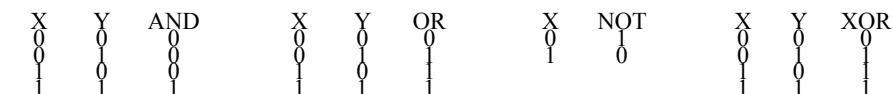
XOR-жоятын немесе екі түрлі мәні болып табылады ма?

Кестені ақиқаттық үшін осы төрт негізгі логикалық операциялар 3.5-суретте көрсетілген.

Бірнеше қысқаша екілік мысалдарды немесе бульдік операцияларды қолдануды қарастырайық.

Біріншіден, қарапайым мысалдармен 1-бит-ті қарастырамыз.

0 OR 0 = 0	0 OR 1 = 1	1 OR 1 = 1
0 XOR 0 = 0	0 XOR 1 = 1	1 XOR 1 = 0
0 AND 0 = 0	0 AND 1 = 0	1 AND 1 = 0



Сурет 3.5 Шындықтың базалық кестесі

Біз екі битті және тиісті ақиқаттық кесте мәнін алайық. 0 XOR 1 көру жолын қарайық, онда $X = 0$ және $Y = 1$.

Тіпті егер біз AND, OR және XOR екі бит (1) алсақ, бұл көп қызықты және пайдалы үлкен бинарлық операцияларды көруге болады. Мысалы олар 11110000 және 10101010 болып келеді. Бұл жағдайда бинарлық операция биттік тәсілімен қолданылады. Яғни, біз бірінші битті әрбір екілік сандарды қолдануға, ал одан кейін екінші бит әрбір екілік сандардан кейін жазады. Бұл жағдайда 8-бит бар, өйткені әрбір санының ұзындығы 8 бит. Операция орындалғанда оңай болуы мүмкін, қолыңыздың екі санын, ал содан кейін операцияларды бағандар бойынша қолдану.

11110000 және 10101010 есептейміз

			1	1		1		0		0		0
0												
және	1		0		1		0		1		0	
				1	0		1	0		0		0
0												

Шеткі сол жақ биттен бастап 1 және 1-1-ге тең. Келесі бағанда оңға қарай қозғалады, ол 1 және 0-ге тең. Әріқарай 1 және 1 1. Сондай-ақ 1 және 0 0. Сонымен, жоғарғы санына назар аударыңыз, біз оңға қозғаламыз, қалған бит бөлігі 0-ге тең, сондықтан соңғы 4 битте барлығы 0.

10101111 00111100 XOR.

			1	0		1		0		1		1
1												
XOR	0		0		1		1		1		0	
				1	0		0		1		0	
1												

Егер екі бит бірдей болса, онда нәтиже 0 болады және егер екі бит ерекшеленсе нәтижесі 1-ге тең. Аса күрделі операциялардың қолданылуы мүмкін, мысалы, (10110011) және (11110000 11001100 XOR). Бірінші-

ден, біз XOR қолдансақ:

			1	1		1	1	0	0	0
0										
XOR	1		1	0		0	1	1	0	0
				0	0		1	1	1	1
0										0

Бұдан әрі, біз (10110011) қолданамыз. Бұл жай ғана битті бізге 01001100 аударады. Ақырында,

				0	1		0	0	1	1	0
0											
және	0		0		1		1	1	0	0	
					0	0		0	0	1	1
0											0

1. (10110011) және (11110000 XOR 11001100) = 00001100.

Біз неге осындай қарапайым операциялармен жүреміз деп ойлайсыз? Ең базалық деңгейде бұл төрт операция барлық компьютер осылайша есептеулерді пайдаланады, біз бұл төрт операцияны орындау үшін тапсырмаларды қосып, азайтып, көбейтіп, бөліп салыстырамыз. Қалай?

Бұрын бұл сұраққа жауап беру үшін, екілік санды қосу арқылы орындап, әдеттегі қосу + операторын қарастырайық. 0111 + 0110 жасайық. Біз бұл қосымшаны орындау үшін, осылайша екі ондық мәндерін қосамыз. Біз айта кетейік, олардың - бағандар бойынша оңнан солға түрінде жазылуы, ал қажет болған жағдайда, енгізу блогын келесі сол жақты бағанада құру. Дегенмен, тек екілік сандар бізге рұқсат 0, 1 тең және біздің бірлігімізде 10 бірлік, бірақ екеуі жауапты болады.

Енді шешіміге келетін болсақ 0111 + 0110. Шешім 3.6 суретте көрсетілген. Соңғы оң жақ бағаннан бастайық. Бұл бағанды ауыстыру үшін (дәлірек айтқанда, 0 көшіру) осылайша, біз 0 (көшіру) + 1 + 0 қосамыз (шеткі оң сандар біздің екі сандарымыз). 0 + 1 + 0 дегеніміз не? Бұл-1. Осылайша, біз сомасын жазамыз, 1 ауыстырылмайды. Келесі бағанда біз 0 (ауысу) + 1 + 1 0 + 1 + 1 тең 2. Дегенмен, біз екіні төмен жазуға рұқсат бермедік.

Жалғыз сандар рұқсат етілген 0 және 1. Екілік жүйе дегеніміз не? 102. Бұл 6, біз 0 жаздық (соңғы оң сан 102) және 1 көтереді. Естеріңізге сала кетейік, бұл ауыстыру дәрежесі 2 болып табылады, сондықтан блогы 2,

10 болады. Дегенмен, біз жазуды көшіреміз, 1 разряд бойынша келесі бағанда көрсетілген. Енді үшінші-оң жақ бағанда, 1 (көшіру) + 1 + 1. Бұл 3. Тағыда 3 жазылады. Бірақ 3 екілік жүйеде 112. Осылайша, біз 1 мойнына 1 жазамыз. "Шеткі сол жақ бағанда, біз 1 (көшіру) + 0 + 0 = 1. Осылайша, біз 1 жазамыз және ауыстырмаймыз. Ауысу шеткі сол жақтағы бағаннан шығуы мүмкін, жай-күйі жазылатын жалаулар (бұл тарау битті ауыстыру. Бұл пайдалы болуы мүмкін.. Яғни, бұл екі саны тым үлкен, мақсаттағы орнына кірісіп кетуі (тіркелуі немесе жады).

3.6 суретке назар аударыңыз, бұл үшін әр бағанның екі нәрсесін есептейміз. Үш санын есептеуге (көшіруді қоса алғанда) және жүзеге асыруға, қабылдауға қажетті бағанында қол жүгінеді. Біз үш санды қосу үшін осы бағандағы екі санының және көшірудің екі түрлі сомасының нәтижесін есептейді және жүзеге асырады. Бір баған ішінде көшіру айналады. Мынадай бағанда өткіземіз:

Алгоритм сомасын былайша білдіруге болады:

1. екі екілік сандарды калыптастыру.
2. бағанда қиыр құқығымен әрқашан 0 ішінде ауыстыру.

Қиыр:		1	1	0	0	Алғашқы
X:		0	1	1	1	қиыр 0
Y:	+	0	1	1	0	
Сума:		1	1	0	1	
			Қиыр	Қиыр	Қиыры жоқ	

Сомасы: 1 1 0 1

Сурет 3.6 Екілік санды қосу, көрсету және жүзеге асыру / кию

1. Жұмыс оңнан солға қарай, келесі бағанамен қайталау.
 2. Есептеу сомасы екі екілік санды жеткізуді қосу, ол былайша: 0: 0 барлығы үш
 - 1: 1 бір болып табылады
 - Екі-үш болып табылады 1: 0 (1 + 1 + 0 = 2 немесе 1, сомасы 0, ауысу 1)
 - Үш 1: 1 (1 + 1 + 1 = 3, немесе 11, сомасы 1, ауысу 1)
 - 1.Формула екі екілік санды жеткізуді қосып көшіру, ол былайша болады: 0: 0 барлығы үш
 - 1: 0 бір болып табылады , екі, үш 1: 1 барлық үш 1: 1
 3. Осы бағанда көшіру болып табылады.
- Біз айта кетейік 1001 + 0011. Біріншіден, кіріс нөмірлері:

$$\begin{array}{r} 1001 \\ + 0001 \\ \hline \end{array}$$

Шеткі оң жақ бағанасында ауыстыруға арналған, бұл 0. Бұл 0, 1, 1. Сатысы жоғары 3 а сәйкес, екі 1 бит сомасы 0 береді. 3b қадамына сәйкес, екі 1 бит бізге 1 ауыстыруға береді. Осылайша,

$$\begin{array}{r} \text{Қиыр: } ---10 \\ 1001 \\ + 0011 \\ \hline ---0 \end{array}$$

Қазір екінші бағанда (оң жақта) 1 көтеруге, ал екінші сан: 0 және 1, сондықтан бізде екі 1 бит. Бұл сомасы 0 және көшіру 1. Енді біздің ішінара нәтижесі мынадай.

$$\begin{array}{r} \text{Қиыр: } -110 \\ 1001 \\ + 0011 \\ \hline -00 \end{array}$$

үшінші баған, біз 1 және екі санды 0-ге ауыстырамыз. Бұл бізге 1 және көшіру 0. Енді біздің ішінара нәтижесі төмендегідей:

$$\begin{array}{r} \text{Қиыр: } 0110 \\ 1001 \\ + 0011 \\ \hline -100 \end{array}$$

шеткі сол жақ бағаннан бастап 0 сандар 1 және 0 ауыстыруға аламыз. Бұл бізге 1 және 0 көшіруге мүмкіндік береді. Біздің қорытынды сомасы 1100. Егер біз деци-Мәлға мәселесін түрлендірсек, онда біз $9 + 3 = 12$ орындайтын едік.

Бірнеше дербесті қарап көріңіз.

$$\begin{array}{l} 0001 + 0101 = 0110 \\ 1001 + 0011 = 1100 \\ 0111 + 0011 = 1010 \end{array}$$

Енді, біз бұл алгоритмді компьютерге қосамыз, өйткені бізде тек AND, OR, NOT және XOR операциясын қолдану керек пе? Біз алгоритмді қосу есептеулерді екіге бөлу керекпіз, яғни есептеу сомасы және есептеуді ауыстыру. Біз бастауға тиіспіз, өйткені ол сәл жеңілдірек түсінік. Естеріңізге сала кетейік, үстінен жүзеге асыру болып табылады: 1. егер екі немесе барлық үш сандар 1. үш мәндері X, Y және C (алып жүру үшін). Төрт ықтимал құрамалары бар, бұл ретте екі немесе үш 1s . Бұл $X = 1$ және $Y = 1$, $X = 1$ және $C = 1$, $Y = 1$ және $C = 1$ немесе $X = 1$, $Y = 1$ және $C = 1$. Шын мәнінде, біз төртінші мүмкіндіктерді ойламаймыз, өйткені басқа үш шындық деп салыстырамыз. Осылайша, біз көшіруді анықтаймыз, осының көмегімен мынадай булелді логика шығады: (X AND

Y) OR (X) NEMECE (Y және C). Есептеу үшін біз білуіміз тиіс, егер бар болса, биттердің ішінде 1 үш бит (X, Y, C) немесе үш 1s үш бит. Біз ақылды ілінісумен бірге екі XOR операцияны анықтаймыз:

$$(X \text{ XOR } Y) \text{ XOR } C$$

Boolean операцияларын пайдалана отырып, біз мысалды 3.6 суретте қайта жасайық. Екі нөмірлері 0111 + 0110. Шеткі оң жақ бағаннан бастап, біз $X = 1$, $Y = 0$, $C = 0$ сома $(X \text{ XOR } Y) \text{ XOR } C = (1 \text{ XOR } 0) \text{ XOR } 0 = 1 \text{ XOR } 0 = 1$. Осылайша, бізде 1сомасын көшіру болып табылады (X және Y немесе X және C) немесе $(Y) = (1 \text{ және } 0) \text{ NEMECE } (1 \text{ және } 0)$ немесе $(0, 0) = 0 \text{ NEMECE}$

0 немесе $0 = 0$. Осылайша, шеткі оң жақ бағанда, біз 1 сомасын және 0 көшіреміз.

Екінші оңға бағанында, біз $X = 1$, $Y = 1$, $C = 0$. Біздің сомасы $(1 \text{ XOR } 1) \text{ XOR } 0 = 0 \text{ XOR } 0 = 0$. Біздің көшіру білдіреді $(1, 1) \text{ OR } (1 \text{ және } 0) \text{ NEMECE } (1 \text{ және } 0) = 1 \text{ NEMECE } 0 \text{ OR } 0 = 1$.

Осылайша, біз сомасы 0 және ауыстыру 1

Үшінші оң бағанында біз $X = 1$, $Y = 1$, $C = 1$. Біздің сомасы $(1 \text{ XOR } 1) \text{ XOR } 1 = 0 \text{ XOR } 1 = 1$. Біздің көшіру білдіреді $(1, 1) \text{ OR } (1, 1)$, NEMECE $(1, 1) = 1$ немесе 1 немесе $1 = 1$.

Осылайша, біз 1 сомамыз және 1 көшіру бар .

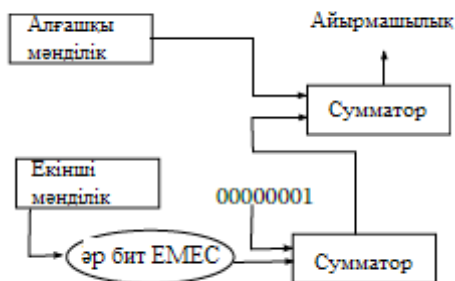
Үшінші сол жақ бағанда $X = 0$, $Y = 0$, $C = 1$ бар. Біздің сомасы $(0 \text{ XOR } 0) \text{ XOR } 1 = 0 \text{ XOR } 1 = 1$ және біздің көшіру $(0 \text{ AND } 0) \text{ NEMECE } (0 \text{ және } 1) \text{ NEMECE } (0, 1) = 0$ немесе 0 немесе $0 = 0$ болып табылады. Осылайша, біздің шеткі сол жақтағы бағанда бар 1 сомасын 0 жүргізеді.

Ескере отырып, біз екі екілік санын қосу көмегімен AND, OR және XOR орындаймыз. Қалай есептейміз? Мысалы, бізде екі санының X және Y және біз есептеу үшін $X - Y$ аламыз. Сіз алгебрадан есіңізге түсірсеңіз $X - Y = X + (-Y)$. X және Y екі комплемент нөмірін байланыстыру реакциясы ұсынылды, біз өзгертеміз Y болады $-Y$. Біз оң теріс санды түрлендіреміз немесе теріс оң сан түріне дейін толықтырамыз. Біз барлық 1 битті қосамыз. Біз барлық соқаларды бірқатар ауыстырдық па? Біз NOT операциясын осылайша шын мәнінде есептедік $X - Y = X + ((\text{ }) + 1)$. Осылайша, бір азайтуды және сол операциялар пайдалана алады, бұл үшін жоғарыда сомасын алып жүреді, бірақ біз айта кетейік, оған операция ЕМЕС, ең алдымен Y , содан кейін қосу 1 емес Y . 1 рет пайдалана отырып қосамыз, сомасы мен операцияларды орындаймыз. Сурет 3.7.

Көбейту сәл қиынырақ, бірақ тек бірнеше тармақ өзгертілді ($x * y$ і қосу x бірге y ретін талап етеді). Бөлім сериясы есептенеді (шегеру y x , біз жетеміз 0 немесе теріс x / y талап етеді). Осылайша, біз алгоритмдерді қосуды және азайтуды қолдануға, жоғарыда сипатталған көбейтуді және бөлуді бірнеше рет орындайды. Олар комбинациясын AND, OR, XOR және NOT қолданады.

Қалай салыстыруға болады? Шын мәнінде, бұл өте оңай. Салыстыру үшін екі мәндерді X және Y , біз $X - Y$ қарап, нәтиже оң болады (бұл жағдайда $X > Y$), теріс (бұл жағдайда, $X < Y$), немесе нөл (бұл жағдайда $X = Y$). Тексеру үшін теріс, қараңыз шеткі сол жақтағы бит айырмасы ($X - Y$); егер ол 1-ге тең болса, онда айырмасы теріс (естеріңізге сала кетейік, толықтыру түрінде екі, шеткі сол жақтағы бит белгісі болып табылады), бұл $X < Y$. Тексеру үшін нөлге НЕМЕСЕ барлық соқалар бірге қарап, нәтиже 0-ге тең болады, яғни $X - Y = 0$ немесе $X = Y$. Мысалы, егер біз 00000001 немесе осы бит 1 тең, сондықтан нөлге тең маңызды емес. Оңды мәнінде ең қиын анықтау, біздің нәтижеміз нөлге тең болмаса және бұл шеткі сол жақтағы бит 0 тең. Алайда, оң сан болса анықтау болып табылады, бірде теріс, бірде нөлге тең ! Егер айырма оң болса, онда $X - Y > 0$ немесе $X > Y$.

Қалай салыстырмаймыз? Әдетте, басқа сандар тек басқа да заттар, біз бұл рәміздер мен жолдың таңбасын салыстырамыз. "Франк" және "Фред" қайсысы артық екенін көруді біз рәміздерді салыстырамыз, әзірге біз жолдың соңына қол жеткіземіз (бұл жағдайда жолға тең) немесе бізге сәйкес келмейді. Мұнда біз сәйкессіздік болса табамыз.



Сурет 3.7 екі санның екі толықтырулармен және NOT азайту.

Үшінші символ 'a' 'e'-ге қарсы. Біз 'a' - 'e' қалай есептейміз? Естеріңізге сала кетейік, рәміздер жадында пайдалана отырып, ASCII ұсынуға сақталады. ASCII 'a' 96 'e' 100-ге тең. Сондықтан "Франк" пен "Фред"-ты салыстырамыз, біз 'F' - 'F', сондықтан біз 0 жасаймыз, 'R', - 'r' 0-ге тең, сондықтан біз 'A' - 'e' жасаймыз, ол -4 болып табылады. Өйткені 'a' < 'e', "Фрэнк" < "Фред".

Жұлдыздардан жолдау

Физик Франк Drake жерден тыс ақыл іздеудің негізін қалады (SETI). Шамамен 1961 жылы жұмыс басталды, радиотелескоптар (және басқа да бақылау нысандары) жұлдыздардың хабарын тыңдау үшін. Басқа планеталықтан қандай хабарлама нысанды алуы мүмкін? Дегенмен, біз ешбір түсінуге ұғым жоқ, бұл кез келген кейбір екілікті хабарлама кодталып ұсынылды. Не үшін? Өйткені ақпарат екілік жүйеде оңай беріледі, топтамасын қосу / өшіру импульсі жіберіледі. Бірақ, мұндай хабарлама қалай көрінеді? Шетелдіктер ұғымды таңдау керек, оны қалай оңай түсіндіруге болады. Мысалы, біз ASCII хабарламасын жібереміз, өйткені ASCII адам өнертабысқа қатаң және ол шетелдіктерге мағынасыз болады. Хабарламаны шифрлеп көрсетуі қиын болуы мүмкін, келесі әріптестеріме аталық жағдайда оның мағынасын анықтау Олар істей алмады!

```
11110000101001000011001000000010000010100
10000011001011001111000001100001101000000
```

```

00100000100001000010001010100001000000000
000000000001000100000000000101100000000000
0000000100011101101010101000000000000000
00001001000011101010101000000000101010101
00000000011101010101110101100000001000000
00000000000100000000000001000100111111000
00111010000010110000011100000001000000000
10000000010000000111110000001011000101110
1000000011001011111010111100010011111001
0000000000011111000000101100011111100000
10000011000001100001000011000000011000101
001000111100101111

```

Басқа операция түрі булелді, бұл бізге, біздің компьютерлер орындау үшін белгілі маскалар. Маска салыстырады екі екілік маңызы бар бірге пайдалана отырып, бір AND, OR немесе XOR. Бір пайдалану және маска жабылған үлгілерде көмегімен екілік сандар.

Екілік сандарды қолдану мысалдары

Мұнда біз үш өте әр түрлі тәсілдері, ол екілік санын пайдаланылатын компьютерлер. Бірінші жағдайда, біз қараймыз екілік сандарды пайдаланылатын желілік мекен-жай желілік масканы. Содан кейін, біз кейбір кең таралған видео форматтарын ұсыну файлдарын қарастырамыз. Және, ақыр соңында, біз пайдалану ретінде екілік сандарды орындау үшін анықтау және қателерді түзетулерді қарастырамыз.

Желілік мекен-жайлар және желілік маскалар

Компьютерлік желілер қамтамасыз етеді жақсы үлгісі ретінде қолданылуы мүмкін бинарлық ұғымдар білдіңіз, әлі күнге дейін. Әрбір компьютер желісі болуы тиіс бірегей желілік мекен-жайы. IPv4 хаттамасы бойынша Интернет нұсқасы 4) адрестеу нүктелік ондық форматтағы осы мекен-жайды пайдаланады.

IPv4-мекен-жайы төрт сандар, әрбір октета деп аталады. Әрбір нөмірі тұрады белгінің 8-разрядтық екілік сандарды сақтай аласыз. Естеріңізге сала кетейік, 8 бит, сіз кез келген ондық саны 0-ден 255. Осылайша, IP-мекен-жайы, яғни төрт санының 0-ден 255, әрбір саны бөлінген нүктесі (кезең). Мысалы, IP мекен-жайы компьютер болуы мүмкін 10.251.136.253. Кез-келген желілік мекен-жайы, жұмыс оңай түрінде ұсынылған бір 32-разрядтық екілік сандар. Біз оны, 4 октета үшін оны оңай әрі қарай ұшырамыз.

IP-мекен-жайы топтастырылған осылайша, бұл компьютерлер октеты кейбір желілік қорында. Бұл, меніңше, онда көше мекен-жайлары топтастырылған пошталық кодтар болады. Компьютерлер бір желі, мүмкін бір-бірімен өзара іс қажеттілігінсіз бағдаршаның жұмысын атқарады. Бағдаршамалар қолданылады жеңілдету үшін байланыс желілерімен жүзеге асады. Бұл, меніңше, онда пошта бөлімшелері жеңілдету үшін поштаны жеткізу әр түрлі пошта индекстері пайдаланылады. Сіз осы пошта бөлімшесіне қандай көше мекен-жайлары тиесілі екенін, бұл мекен-жайлардың пошталық индексін айта аласыз. Компьютерлердің ұқсас тұжырымдамасы. Жасырын шегінде сіздің IP-мекен-жайы болып табылады өз кезегінде пошталық индексі, мекен-жайы желісі. Біз бұл ретте желілік мекен-жайды атаймыз.

Анықтау үшін, желілік мекен-жайы IP-мекен-жайына жатады, компьютерлер бетперде желісі. Көрейік, мысалы сіздің IP-мекен-жайыңыз 10.251.136.253, және сіздің бетпердесі болып табылады 255.255.255.0. Екілік ұсыну (нүктелержеңілдетуүшіноқулары) осы сандардың:

IP-мекен-жайы :00001010.11111011.10001000.11111101
желілік бетпердесі: 11111111.11111111.11111111.00000000

Есептеу үшін мекен-жайы, желі, екілік және операция орындалады, арасындағы осы екі 32-биттік санының желілік мекен-жайы әкеледі:

Желілік мекен-жайы: 00001010.11111011.10001000.00000000
(10.251.136.0)

Естеріңізге сала кетейік, екілік және 1-і болып табылады, тек егер екілік бит болып табылады 1. Бұл бетпердесі бар нәтижесі оралған алғашқы үш октета бүкіл олардың толықтығы, өйткені олар қазіргі уақытта операция AND отырып, 11111111, түпкілікті октета жасырынады болып табылады, немесе нөлге айналды, өйткені бұл октет операция кезінде AND 00000000 басталады.

Келтірілген жоғарыда мысал ретінде, біз маскарларды 255.255.255.0 келтіреміз. Бұл маска таңдалды, өйткені барлық компьютерлер осы желінің ортақ алғашқы үш октетасы. Яғни, мекен-жайы желісінің алғашқы үш октета төрт октеттер. Мекен-жайы және әрбір құрылғы желісіне түпкілікті болып табылады октет, осылайша бере отырып, әрбір құрылғы бірегей мекен-жайды белгілейді. Енді, бетперденің 255.255.255.0 256 (28) ықтимал мекен-жайларының желісі бар. Осылайша, бұл ұйым болар

еді, 256 бірегей IP-мекен-жайларды ұсынады. "Нүктелі кондық форматында, желілік мекен-жайы оралып, осы операциялар 10.251.136.0. Бұл, желілік мекен-жайы диапазонында 10.251.136.0 арқылы 10.251.136.255, мекен-жайын қоса алғанда, біздің бастауыш құрылғыны 10.251.136.253 қамтамасыз етеді.

Екі компьютермен бірге бір желілік мекен-жайы алады, өзара іс-қимыл бір-бірімен қажеттілігі жоқ бағдаршамалар пайдаланады. Екі компьютердің әр түрлі желілік мекен-жайлары білу, олар байланысу өз бірлік төңірегіндегі бағдаршамалар (бағдарша әдеп кіретінде жиі аталады) үшін қарым-қатынас екі арасындағы құрылғылармен жабдықталады.

Бұл ұғымды түсіну оңай, қашан сіздің бетперденіз тері біркелкі шекараларымен октет болғанда (яғни, нүктесі), бұл біздің мысалымызда жоғары. Дегенмен, тағы бір мысал қарастырайық, онда ол емес. Біз бір IP-мекен-жайды пайдаланатын боламыз, 10.251.136.253, бірақ маскасы 255.255.240.0 болады. Есептеу үшін біздің желілік мекен-жайы, біз екілік осы екі сандар керек. Сурет 3.8 көрсетеді қолдану NETMASK 255.255.240.0, онда біздің желілік мекен-жайы көрсетілмейді 10.251.128.0, 10.251.136.0 бетперде желі санымен бит әріпі құрайтын желісі бөлігі жұмыс мекен-жайымен айқындалады. Кейбір желілік мекен-жайлары құрамында тек бірінші октет болады. Мысалы, бір желі 95.0.0.0 (немесе 01011111.00000000.00000000.00000000) болуы мүмкін. Бұл желі ішкі желі бүркеніштің бірі 255.0.0.0 пайдалануға болады.

Әлбетте, бұны түсіну екілік сандарды және олардың операциялары үшін өте маңызды, оны кім зерттейді. Желілік мамандар жетік меңгеруі тиіс түсіну логикалық операциялар AND, OR, NOT және XOR. Біз желісін 12-тарауда сипаттаймыз.

Сурет файлдары

Бұрын осы тарауда, бұл сандарды біз көрдік (оң, теріс, бүтін, бөлшек оң сан) сақталуы мүмкін " екілік түрінде не тікелей, не арқылы беру сияқты қосымша немесе өзгермелі нүктесі. Сондай-ақ, бұл рәміздер ұсынылған екілік жүйеде пайдалана отырып, кодты атты ASCII біз көрдік. Ретінде суреттер, компьютерде сақталған? Ең қарапайым белгілі растрлы сурет формат ретінде.

Кез-келген суретті жарық нүктелерінің жиынтығы білдіреді. Әрбір жеке нүктесі пиксель деп аталады.

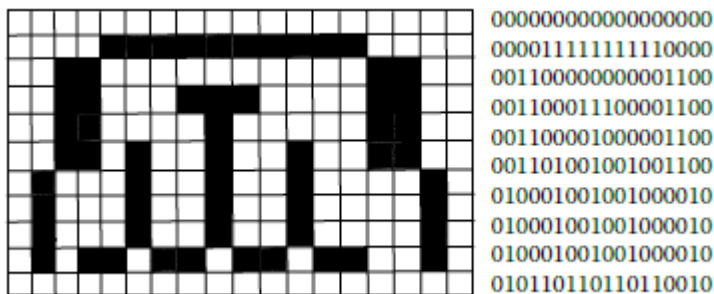
(Сурет элементі). Қара-ақ түсті бейнесі, пиксель болады өзіне бір екі түсті, қара немесе ақ. Біз елестете отырып, осы екі түсті пайдалана отырып, 1S және 0s (әдетте, ақ, 1 қара тең 0). Сақтау үшін қара және ақ болып табылады, мәселе орналастыру файлды әрбір пиксель, жол наси-

хаттайды. Сурет 3.9-суретте қарапайым нүктелік суретті білдіреді. Фигурасы да, бейнесі және сақтауға екілікті (оң жақтағы сурет) қамтиды. Үшін қара және ақ $X \times Y$ пиксель (X жолдарының, Y -бағандар), сақтау талап етіледі $X * Y$ бит. 1000×1000 пикселдер сурет 1000000 бит қажет. Бұл 125000 байт, немесе шамамен 125 КБ болар еді. Түрі қара-ақ түсті бейнесі сұр түсті сурет болып табылады. Мұнда әр түрлі реңктері сұр қатаң қара және ақ қосылады. Ұсыну үшін сурет, ұқсас қара-ақ түсті бейнесі, әрбір пиксель сақталады, түрінде бір, бірақ қазір мәні орналасқан диапазонында ең төменгі қарқындылығы (қара) қарқындылығының (ақ) маңызы бар. Кез келген мәні арасындағы түсі болып табылады.

Айпп: 00001010 . 11111011 . 10001000 . 11111101 (10.251.136.253)
Желілік: 11111111 . 11111111 . 11111000 . 00000000 (255.255.240.0)

Желілік
мекен-жай: 00001010 . 11111011 . 10000000 . 00000000 (10.251.128.0)

Сурет 3.8 Ішкі желісінде қолдану маскалары.



Сурет 3.9. Ақ-қара нүктелі сурет және тиісті екілік файл.

Көбінесе, қарқындылық деңгейі 0 ден 255 санымен болады, сондықтан 1 байтта әрбір пиксель сақталуы мүмкін. 1000×1000 пикселдер суреті 1000000 байт қажет (шамамен 1 МБ). Қара-ақ сурет 8 есе мөлшерінен асып кетеді, өйткені біз оның мағынасын ұлғайтып, талапты сақтау 1 бит пиксельден 1 байтқа дейін арналған. Компьютерде шамасымен қызыл, жасыл және көк түстер ұсынылған, осы пикселдің жарығын құрайды. Біз бұл стратегияны RGB сияқты деп атаймыз. Түрлі-түсті сурет-

терді алу үшін, біз әрбір пиксельді сандарды триосы ретінде сақтаймыз, қызыл санды, жасыл түсті санды және көк түсті санды. Жарты тондық суреттерді біз әрқайсысын, әр интенсивті түріндегі санды 1 байтта сақтаймыз, бізге 0-255 диапазоны не береді. (0, 0, 0) түсі ұсынылды, яғни қызыл, жасыл, көк емес, қара. (255, 255, 255) түсі ұсынылды. (255, 0, 255) түсі ұсынылды, күлгін (толық қызыл, жасыл, толық көк). " Екілік түрінде күлгін болады, үш санының (11111111, 00000000, 11111111). Ыңғайлы болу үшін RGB маңызы он алтылық орнында жиі білдіреді. Сондықтан FF00FF күлгін. Бұл жазу веб-беттер өңдеуі пайдаланылады. Мысалы сары, барлық жасыл және көк, қызыл немесе (0, 255, 255) немесе 00FFFF.

Мәні 884400 болады (128, 64, 00), яғни жартысы қызыл және төрттен бірі жасыл, қоңыр -қызылт сары түс болады.

RGB үшін 1000×1000 пикселдер суреттерінде 3000000 байт немесе шамамен 3 МБ қажет. Назар аударыңыз, RGB көмегімен ұсыну ретінде, бұл әрбір пикселдегі бейне 16 миллион түстен астам болуы мүмкін ($256 * 256 * 256$). Қара-ақ түстен одан сайын арттыру файл өлшеміне өтуді сіз көре аласыз. Файл өлшемі суреттерді азайту үшін қандай тәсілдері бар? Иә, файлдарды қысу көмегімен. Кең таралған екі нысандарын қысу болып табылады, олар GIF және JPG. GIF Graphics Interchange Format ретінде білдіреді. Идея мынада: әдетте адамдар миллиондаған түстерді кез келген нақты жолдарда ажыратуы мүмкін емес. Көп орынды неге сақтау үшін пайдаланады, бір қызыл түсті көрінуі мүмкін бе? GIF-те түс палитрасы болған жағдайда, 256 ең танымал түстен тұрады. Әрбір пиксель сілтемесін түс палитрасында пайдалануға мүмкіндік береді, тек 1 байт орнына 3-ші байт (RGB үшін талап етіледі); осылайша, мөлшері түрлі-түсті суреттер 3 коэффициенті азаяды. 1000×1000 пикселдер түрлі-түсті бейнесі тек 1 МБ дискілік кеңістікте талап етеді. Өкінішке орай, егер сурет бастапқы түсті пайдаланса, олардың бір бөлігі болып табылады, онда суреттер әртүрлі көрінеді, ал кейбір жағдайларда сіз дұрыс емес *grayscale* – түсті аласыз, мысалы сары түстің орнына көк ақшыл түсті аласыз .

GIF-ке жиі қарағанда JPG формат. JPG Joint Photographic ретінде білдіреді. Experts Group (түпнұсқа аббревиатура үшін JPG файл .jpeg).

Қол жетімді палитрасында нәтижелендірілген файл GIF BMP файл бірдей көрінеді және сондықтан белгілі ретінде қысу шығынсыз болады. Бірақ JPG қысу шығынмен пайдаланады. Айырмашылық мынада, суреттерді қысқаннан кейін біраз сапасы жоғалады. Сіз мөлшерді сақтау үшін сурет сапасын жоғалтасыз. Сығу форматы JPG пикселді біріктіру арқылы жұмыс істейді, кейбір (нүкте) топтардың пикселі (нүкте) бір-

ге одақтасады. Шын мәнінде, сурет кейбір рұқсатты жоғалтады, оның біразы су ойып кеткенге айналады. Сонымен қатар, егер суреттер түрлі-түсті болса, ол RGB-да баламалы форматы түсте ауыстырылады, белгілі YUV сияқты, онда әрбір пиксельді пайдалана отырып, жарықтығы (Y) және люминесценция (U, V) ұсынылады. Бұл үлгі екі немесе төрт есе аз орынға қарағанда RGB талап етеді. Сонымен қатар, бейнелері көп JPG қысу шығынмен мөлшерін азайта алады.

Соңғы PNG (Portable Network Graphics) GIF форматын кеңейтеді. GIF сияқты бұл шығынсыз нысанын қысу, ол биттікті пайдаланады. Ол 24-биттік RGB-палитраны қолдайды, сұр градациясы және бейнелердің палитралары жоқ. Және де JPG суретті азайтып қысады, бірақ шығынсыз алгоритмді пайдалана отырып, қысу бейнесі бірдей сапасын білдіреді. JPG, PNG қарағанда қысуы мүмкін немесе мүмкін емес, шын мәнінде мөлшерін азайту файлы.

Қателерді түзету және анықтау

Бір маңызды міндеті компьютермен жұмыс істеген кезде сақталған ақпаратты өңдеу кезінде бір орыннан басқа орынға деректерді өткізу. Бұл екіталай, бірақ қателер болған жағдайда ауыстырылады және бір компоненттің басқа компьютерде деректердің пайда болуы, бірақ бұл әлдеқайда күтуге болады, бұл қателер процесі беру кезінде туындауы мүмкін. Осылайша, кейбір нысанын анықтау және қателерді түзетудің көп болғаны дұрыс.

Ең қарапайым нысаны шындықта қателерді анықтап пайдалану арқылы. Жалпы, бақылау шындығы теңдікті білдіреді. Бірақ компьютерді талқылау кезінде, тақ бірқатар шындыққа жатады (немесе жұптық емес). Атап айтқанда, біз табысын 1 бит реттілікпен санауды деректерді көруге, егер бұл есептегіш жұп немесе тақ болып табылса. Содан кейін біз бұл мүмкіндікті пайдаланып қарапайым қателерді беру кезінде табу.

Бит шындығын әрбір байтқа қосамыз (8 бит). Бит шындығы тең болса, 1-саны 1 бит тақ, егер саны 1 бит бәйте тақ болса бит шындығы 0-ге тең болады. Яғни, 8 бит плюс тақ бит әрқашан тақ саны 1 бит болуы тиіс. Бұл стратегия ретінде белгілі тақты (бар вариация деп аталады тақты емес, онда 8 бит плюс бит әрқашан тақ санды 1 бит болуы тиіс).

Тіпті біз паритет былайша пайдаланатын боламыз. Ескере отырып, байт, бұған дейін ИГД жылжиды, біз тақ битті генерациялаймыз. Енді бізде 9 бит. Тапсыру / жылжыту 9 бит. Алушы 9 бит. Егер 9 бит жұптық бар болса, онда қателер жоқ, қателікте ИГД лақтырылады, осы шама наразылықта болу керек екенін тұтынушы сұрайды.

Біз жұп немесе тақ сағаттарды қалай анықтаймыз? Бұл өте оңай. Жай XOR операциясы көмегімен (сурет 3.5). XOR операциясы ғана 2 бит бір рет қолданылады, бірақ біз XOR операциясы тізбектерімен бірге істей аламыз. Осылайша, 8-биттік деректер, біз XOR бірінші екі биті, содан кейін XOR нәтижесін үшінші битпен, содан кейін XOR нәтижесін төртінші бит және осылайша бұдан әрі. Кейбір мысалдарда есептеулерді қолдана отырып тақты битті қарастырайық. Біз тірек нүктесін 10010011 басқа орынға компьютерде немесе желіде жылжытамыз. Бірінші қадам тақ битті есептеуі тиіс:

$$\begin{aligned}
 & (((((((1 \text{ XOR } 0) \text{ XOR } 0) \text{ XOR } 1) \text{ XOR } 0) \text{ XOR } 0) \text{ XOR } 1) \text{ XOR } 1) \text{ XOR } 1) = \\
 & ((((((1 \text{ XOR } 0) \text{ XOR } 1) \text{ XOR } 0) \text{ XOR } 0) \text{ XOR } 1) \text{ XOR } 1) \text{ XOR } 1) = \\
 & ((((((1 \text{ XOR } 1) \text{ XOR } 0) \text{ XOR } 0) \text{ XOR } 1) \text{ XOR } 1) \text{ XOR } 1) = \\
 & (((((0 \text{ XOR } 0) \text{ XOR } 0) \text{ XOR } 1) \text{ XOR } 1) \text{ XOR } 1) = \\
 & (((((0 \text{ XOR } 0) \text{ XOR } 1) \text{ XOR } 1) \text{ XOR } 1) = \\
 & (((0 \text{ XOR } 1) \text{ XOR } 1) \text{ XOR } 1) = \\
 & ((1 \text{ XOR } 1) \text{ XOR } 1) = 0
 \end{aligned}$$

Сондықтан 10010011 тақты 0 бит. Бұның мағынасы бар, себебі жалпы жұп саны 1 бит. 10010011 төртте 1 бит бар және осылайша сақтау үшін 1 бит санын тіпті 0 қосамыз. Енді 9 бит төрттен 1s және 0s тұрады. Бұл 9 битті және алушы 10010011 & 0 береді. Алушы ұқсас XORing 9 битті операциясын және 0 желдерді орындайды. Сондықтан, дұрысын яғни, ешқандай қателер туындамаған. Енді қарастырайық, егер 9 битті алу кезінде 10010011 & 1 болды. Бұл жағдайды болдырмау НЕМЕСЕ есептеу:

$$((((((((((1 \text{ XOR } 0) \text{ болдырмайтын НЕМЕСЕ } 0) \text{ болдырмайтын НЕМЕСЕ } 1) \text{ болдырмайтын НЕМЕСЕ } 0) \text{ болдырмайтын НЕМЕСЕ } 0) \text{ болдырмайтын НЕМЕСЕ } 0) \text{ болдырмайтын НЕМЕСЕ } 1) \text{ болдырмайтын НЕМЕСЕ } 1) \text{ болдырмайтын НЕМЕСЕ } 1) = 1$$

Осылай қателік туындады. Қай жерде қателік шықты? Бір тақ битті пайдалана отырып, біз білеміз, себебі, қателік бар, алушы лақтырылған деректерді қайта жіберуін сұрайды. Назар аударыңыз, бұл жағдайда туын-

даған қате тақ битті. 10010010 & 0 басқа мысал бола алады, онда бит тақ емес, құрамында қате, бірақ бұл байтта қатесі бар (соңғы бит 1 тең болуы тиіс). Алушы осы 9 битті XOR –ды есептеуші алады

(((((1 XOR 0) болдырмайтын NEMECE 0) болдырмайтын NEMECE 1) болдырмайтын NEMECE 0) болдырмайтын NEMECE 0) болдырмайтын NEMECE 1) болдырмайтын NEMECE 0) болдырмайтын NEMECE 0) = 1

Мынаған назар аударыңыз, бұның қатесін анықтауға тақ бит ғана көмектеспейді, бірақ егер екі қателер пайда болса бізді қалдырмайды. Айталық, 10010000 & 0 алушы. Бұл жағдайда, соңғы 2 бит дұрыс емес.

Егер біз XOR бұл 9 бит, біз 0, бұл қателіктер болған жоқ. Осылайша, бит жұптық табуы мүмкін қателіктер 1 емес, 2 емес, мүмкін қатені түзету. Егер біз дұрыс қателер, біз керек еді қарағанда 1 бит жұпты.

Сонымен қатар, бит тақты, міндетті түрде байланысты болуы байтпен аяқтайды делік. Бұрын осы тарауда, маңызы бар ASCII пайдаланылды 7 бит байтты естеріңізге сала кетейік. Сегізінші бит үшін ақпаратты сақтау туралы жұптық пайдаланылуы мүмкін. Мысалы, "a" әрпі ретінде ұсынылады x1100001. Бастапқыда x пайдаланылады, сондықтан біз, әдетте, орналастыруға 0 қажет. Дегенмен, бит, біз қойған бит тақты онда жұмсайды. Сондықтан, шын мәнінде, 'a' сақталатын болады ретінде 11100001, егер біз қосу үшін бит жұпты болса.

Басқа пайдалану тепе-теңдік жеткізіледі RAID технологиясы. RAID жаңа нысаны болып табылады диск сақтау құрылғылары деректері бар бірнеше дискілердің бір шкафында болады. Идея мынада: қосымша дискілер мүмкін қамтамасыз ету де артық деректерді сақтау және қол жеткізу мүмкіндігі бірнеше диск бір мезгілде жеделдету үшін қолжетімділік саясатын дербес әзірлейді және бекітеді. RAID талқыланады толығырақ " 15-тарауында (ақпараттың сақталуын қамтамасыз ету және қауіпсіздік). Мұнда біз ара-қатынасы ретінде пайдаланылуы мүмкін қамтамасыз ету үшін түзету қателерді қысқаша қарастырайық.

11001111	01010101	11011011	00000011

Біз есептеуге бит бойынша XOR әрқайсысы бойынша 4 байт. Бұл дегеніміз, біз жасаймыз бірінші бит (ең сол жақ бит) 4 байт және есептеу XOR бұл 4 бит. Бұл тақ бит бірінші бит төрт байт. Біз бұл үшін әрбір бит 4 байт.

	11001111	Бірінші байт, сақталады первом диске
XOR	01010101	// Екінші байт, сақталады втором диске
XOR	11011011	// Үшінші байт, сақталады третьем диске
XOR	00000011	Төртінші байт, сақталады

Туралы ақпарат алу үшін тақ, онда біз есептейміз әрбір XOR төмен жүріп жатыр. Бірінші бит тақ $((1 \text{ XOR } 0) \text{ XOR } 1) \text{ XOR } 0 = 0$. Екінші бит тақ формула $((1 \text{ XOR } 1) \text{ XOR } 0) \text{ XOR } 1 = 0$.

Үшінші бит жұпты формула $((0 \text{ XOR } 0) \text{ XOR } 1) \text{ XOR } 1 = 0$. Төртінші бит тақ формула $((0 \text{ XOR } 1) \text{ XOR } 1) \text{ XOR } 0 = 0$. Бесінші бит жұп формула $((1 \text{ XOR } 0) \text{ XOR } 1) \text{ XOR } 0 = 0$. алтыншы бит тақ формула $((1 \text{ XOR } 0) \text{ XOR } 1) \text{ XOR } 1 = 0$.

Жетінші бит жұп формула $((1 \text{ XOR } 0) \text{ XOR } 1) \text{ XOR } 1 = 0$. Жетінші бит жұп формула $((1 \text{ XOR } 0) \text{ XOR } 1) \text{ XOR } 1 = 0$.

1. тақты сегіз битті формула $((1 \text{ XOR } 1) \text{ XOR } 1) \text{ XOR } 1 = 0$. Осылайша, біздің тақты байт 01000010. Бұл байт бесінші дискте сақталады.

Енді делік, ұзақ уақыт бойы, ол екінші диск сәтсіз, мүмкін, бар жаман секторы, немесе мүмкін, дисктің барлық беті зақымдалған шығар. Кез келген жағдайда, біз қалпына келтіру барлық деректер көмегімен үш басқа диск, деректер диск жұбы. Біздің 4 байт жоғарыда болған еді біз келесі қол жетімді деректерге жүгінеміз:

11001111 // бірінші байт сақталады бірінші дискіде
 XXXXXXXX // екінші байт деректер зақымдалған
 11011011 // үшінші байт сақталады үшінші дискіде
 00000011 // төртінші байт сақталады төртінші диск
 01000010 // тақ байт бесінші дискте сақталады.

11001111	// бірінші байт сақталады бірінші дискіде
xxxxxxx	// екінші байт деректер зақымдалған
11011011	// үшінші байт сақталады үшінші дискіде
00000011	// төртінші байт сақталады төртінші диск
01000010	// тақ байт бесінші дискте сақталады

XOR төрт тиісті бит үш байт қосу тақ байт алу үшін жетіспейтін дерек-

тер. Мысалы, пайдалана отырып, бірінші бит әрбір байт бар ($((1 \text{ XOR } 1) \text{ болдырмайтын NEMECSE } 0) \text{ болдырмайтын NEMECSE } 0) = 0$). Осылайша, бірінші бит біздің жетіспейтін ақпаратпен 0-ге тең. Екінші бит болады ($((1 \text{ болдырмайтын NEMECSE } 1) \text{ XOR } 0) \text{ XOR } 1) = 1$). Қараңыз, егер сіз жұмыс істеу арқылы қалған бөлігін осы мысал үшін барлық жетіспейтін нүктесіне байланыстыру.

Пайдалану бір тақ биті "байт" деп те атайды тексеру ретінде, көлденең резервтеу, өйткені тақты бит формула бойынша блок болады. Басқа нысаны көлденең тексеру артықшылық болып табылады бақылау сомасы (тарауда 12 қысқаша аталады). Тақты есептеуді үшін RAID пайдаланылады, кейде тік артық кодты, өйткені есептеу біраз мөлшері байт, бит үшін тақ арқылы орындалады.

Әрі қарай оқу

Ұйымдастырылған кітаптың әрбір компьютерінде екілік көріністерді зерттеуді табуға болады (2-тараудағы Компьютерлер бөлімі мен компьютерлерді құрастыру (орнату) қараңыз). Kindle, Binary Made Simple (R. Barton), қолжетімді Amazon Services Digital мәтіні екілік, ондық, он алтылық үлгі арасындағы түрлендіруді қалай орындау туралы пайдалы кеңестерді ұсынады. жүйені нөмірлеу бойынша кітаптар теория мен есептеулерге мұқият фон беріп, мысалы келесідей түрде ұсынады:

1. Конвей, Дж. Сандар кітабы, Нью-Йорк: Springer, 1995.
2. Guilberg, J. және Hilton, П. Математика: Туғаннан сандарды, Нью-Йорк: Нортон, 1997.
3. Ифра, Ж. сандар жалпы тарихы: компьютердің ежелгіден өнертабысқа дейін. Нью-Йорк: Wiley, 1999.
4. Нивен И., Цукерман, Х. және Монтгомери, Г. Сандарды теориясына кіріспе, Нью-Йорк: Wiley, 1991.

Логикалық операциялар (екілік) және логика, сондай-ақ компьютерлік ұйымдастыру мәтіндерінде қаралады.

Сіз сондай-ақ тақырып бойынша тұтас мәтіндерді таба аласыз. Осы жерде бірнеше мәтіндерді тандау бойынша тізімделген.

5. Браун, Ф. Логикалық Пайымдау: Булді теңдеулер Логикасы. Нью-Йорк: Довер, 2012.
6. Грегг, Дж. және нолей: Булеваны түсіну алгебра, сандық сұлбалар, жиындар логикасы. Piscataway, NJ: Wiley-IEEE Press, 1998.

7. Whitesitt, J. Булева алгебра және оның қосымшалары. Нью-Йорк: Довер, 2010.

12 тарауда көрсетілгендей, компьютерлік желілер туралы егжей-тегжейлі осы тарауда жазылған, осы тарауды IP-мекен-жайы мен желілік маскалар туралы ақпарат алу үшін оқыңыз.

8. Miano, J., Қысылған форматтағы суреттер: JPEG, PNG, GIF, XBM, BMP. Reading, MA: Addison Wesley, 1999.

4. Операциялық жүйе қағидасына кіріспе

2-тарауда біз көрген компьютердің аппараттық құралдарын қарастырамыз. Жеңілдету үшін арасындағы байланыс пайдаланушы мен аппараттық құралдары, компьютерлік жүйелер операциялық жүйесін пайдаланады. Бұл тарауда ұсынады көптеген тұжырымдамалар операциялық жүйелер, олардың көпшілігі сілтемелер осы кітаптың 4 кейінгі тарауларында сипатталған. Атап айтқанда, рөлі басқару процестерін, ресурстарын басқару, интерфейс, қорғау және қауіпсіздік енгізілді мұнда, сондықтан кейінгі тексерулер мүмкін баса назар ретінде, оларды пайдалану үшін операциялық жүйелер Windows және Linux керек. Операциялық жүйе ұғымдары үзілісті болса, онда ауысу контекст, жүктеу процесі, сондай-ақ әкімшілік - шот тор, сондай-ақ, енгізіледі. Бүгін қадамдар үшін қажетті қондырғылар жүйесін жұмыс үшін Windows 7 және Linux Red Hat аяқталады.

Осы тараудың оқыту мақсаты болып табылады

1. Операциялық жүйенің рөлін сипаттаңыз .
2. Командалық қатардың интерфейсін мен графикалық пайдаланушы интерфейсін салыстырыңыз.
3. Арасындағы айырмашылықтар нысандары процестерді басқару.
4. Талқылаңыз үзу және ауысу контекст.
5. Виртуалды жадты сипаттаңыз.
6. Талқылаңыз байланысты мәселелер басқармасы ресурстарын: синхрондау .
7. Жүктеу мен баптандыру процесін сипаттаңыз.
8. Қамтамасыз ету нұсқаулықты қадам Windows 7 және Linux Red Hat орнату.

Операциялық жүйе дегеніміз не?

Ең қарапайым сипаттау ретінде, операциялық жүйе (ОЖ) бағдарлама

дегенді білдіреді. Пайдаланушы-бағдарламалаушыға компьютердің нақты аппаратурасының орнына жұмыс істеуге ыңғайлы болатын кеңейтілген виртуальді машинаны ұсынады. Виртуальді машина – бұл пайдаланушы үшін берілген конфигурацияны нақты бар компьютердің бағдарламалық және аппараттық құралдарымен модельдейтін есептеуіш жүйесі. ОЖ аппараттық машинаны виртуальді машинаға түрлендіретін бағдарламалық қамтама қабаты болып табылады. Виртуальді машина конфигурациясының нақты машина конфигурациясынан айтарлықтай ерекшелігі болуы мүмкін. Әрбір аппараттық ресурсқа онымен тығыз байланысқан қандай да бір бағдарламалық ресурс (мысалы, процессор мен процесс) сәйкес келетінінен назар аударайық.

Бейнелеп айтқанда, ОЖ-нің негізгі функциясын сиқырлаушылық деп есептеуге болады – жүйенің нақты бейнесінен жоғарылау күйге түрленуі. Мысалы, операциялық жүйе бір процессорда бірнеше бағдарламаны бірізгілікте орындау иллюзиясын жасай алады. Нәтижесінде пайдаланушы виртуальді машинаны нақты бар машинадан архитектурасы өте жоғары компьютер ретінде қабылдайды.

Кейбір пайдалы Шарттар

Алдыңғы тарауларда біз компьютердің компоненттерінің кез келген жүйесін жүргіздік. Осы анактаулардың қажеттілігіне қарай жаңа мәліметтермен оралайық. АЖ кейбір терминдерін қосамыз.

Құралдар құралғысы

Компьютердің физикалық компоненттері. Біз сынған бұл компоненттер екі санатта болады: ішкі компоненттері мен перифериялық құрылғылар. Ішкі компоненттері алдын-ала жоғарлайды орналасқан аналық такта, кесек пластик немесе шыны талшықты біріктіретін құрылғы болып. Аналық такта бірнеше розеткалар үшін чиптер және жайғасу тактасы болып табылады. Бір чип процессор, басқа да жады чиптерін құрайды. Процессор; ол компоненті болып табылады, компьютерде орындайтын бағдарлама. Процессор орындайды, яғни белгілі бір Таңдау-орындау циклін, таңдаудың әрбір командасындағы бағдарлама бойынша бір соңғы рет еске алу, орындау және оны CPU іске асыру. Бірнеше нысандары бар: жад кэш-жад (SRAM) және динамикалық ОЗУ және ПЗУ. Кэш жылдам жады DRAM баяу жад, бірақ кэш, әдетте, әлдеқайда қымбат қарағанда, DRAM, сондықтан біздің барлық компьютерлер шағын кэш және көп жады DRAM. ПЗУ тек студенттеріне арналған жад пайдала-

нылады, сақтау үшін ғана бірнеше. а. ғ бағдарламалар, олар ешқашан өзгермейді (мысалы, бағдарлама, жүктеу). Құрсым ақпарат компоненттерінен бір басқа мүмкіндік береді. Кеңейту слоттары қосу картасын кеңейтуге, перифериялық құрылғылармен жабдықталған өздері (мысалы, сымсыз модем) немесе қосылуға перифериялық компьютер артындағы порттар құрылғыларға қосылуға мүмкіндік береді. Кейбір ең перифериялық құрылғылар қатты диск, пернетақта, тышқан және монитор болып табылады. Бәлкім, сіз сондай-ақ адам оқуға тиіс жинақтауыш оптикалық дискілерде, сымсыз модем (немесе, мүмкін, желілік карта), дыбыстық карта және сыртқы динамикасы, принтерді білерсіз.

Бағдарламалық қамтамасыз ету

Бағдарлама жұмыс істейтін компьютерде, аталған бағдарламалық қамтамасыз ету ретінде, олардың жеке компоненттерін компьютер (аппараттық қамтамасыз ету) ажырата алады. Бағдарламалық қамтамасыз ету түрінде электр тогы ағатын компьютер арқылы (немесе түрінде магниттік зарядтардың сақталады диск) өтеді. Бағдарламалық қамтамасыз ету, әдетте, қолданбалы бағдарламалық қамтамасыз етуге (бағдарламалар, біз жүгіреміз, орындау үшін қандай да бір тапсырманы немесе жүйелік бағдарламалық қамтамасыз ету (бағдарламалар, компьютер жұмыс істейді, бұл мәселе ортаға) ретінде жіктеледі. Біз, ең алдымен, бағдарламалық қамтамасыз ету жүйесін, бұл мәтінде біз зерттедік. Екі Linux және Windows операциялық жүйесі көптеген программалардан тұрады. Мысалы, Linux, бағдарламасы бар, деп аталатын Ls, RM, CD, және бұдан әрі, орындалатын орындау кезінде команда командалық жолында бар. Бұл бағдарлама сондай-ақ, кейбір аспаптар графикалық интерфейсі бар сияқты файлдық менеджер. Егер сіз қарасаңыз Linux каталогы / бен / USR / бен / SBIN және / USR / SBIN деп аталатын, сіз көптеген бағдарламаларды ОЖ-де білесіз. Көптеген пайдаланушылар өзара іс-қимылын тікелей операциялық жүйесі арқылы осы бағдарламаларды сирек қолданады. Егер сіз қарасаңыз, / USR / жергілікті / каталогтар сияқты, және кейбір мәжбүрлеу - бағдарламалық қамтамасыз етуді сіз таба аласыз. Қолданбалы бағдарламалық қамтамасыз ету мыналарды қамтиды мәтіндік процессорлар, электрондық кестелер, интернет-браузерлер, және компьютерлік ойындар. Windows операциялық жүйесінде таба қолданбалы бағдарламалық қамтамасыз ету директорліктің Program Files және ОС каталогында Windows болады.

Барлық бағдарламалық қамтамасыз ету компьютерлік бағдарламалау сияқты Java немесе C ++ тілінде жазылады. Бағдарлама жазылған адам

бастапқы кодты бағдарлама немесе бастапқы код деп аталады. Java және C ++ сияқты көрінуі ағылшын тілімен ұштастыра отырып, математикалық символдармен белгілейміз. Компьютер бастапқы код түсінбейді. Осылайша, бағдарламашы тиіс бағдарламасын іске қосу аударма компилятор деп аталады. Компилятор бастапқы код аударды және оның машиналық тіліне қабылдайды. Тілдік нұсқасы машинаның орындалушы бағдарламасы болып табылады. Тағы бір сынып тілдік аудармашы, белгілі аудармашы ретінде орындалады. Аудармашы аздап ерекшеленеді, ол біріктіреді трансляционды кодты орындауға, сондықтан іске қосу түсіндірмелі бағдарламасына көп уақыт кетеді. Өйткені бұл әлдеқайда аз тиімді, біз пайдалануға бейім тәсіл неғұрлым ұсақ бағдарламалар сияқты скриптер қабығының, скриптер, веб-браузері мен веб-сервер сценарийлерін пайдаланамыз. Қалай АТ-студент, сіз бірінші кезекте код жазасыз, ол өзгермелі болады. Тілдер сценарийлерін сценарийлер тілінің қабығы Bash, PHP, Perl, J қамтиды.

Компьютерлік жүйе

Компьютерлік жабдықтар, бағдарламалық қамтамасыз ету, пайдаланушы (лар) мен желілері топтамалары. Біздің ОЖ бағдарламалық компоненттер жинағы. Біз бұл компоненттерді 4 категорияға бөлеміз, ол жұмыстың төменінде сипатталады.

Құрылғы драйвері

Құрылғы драйвері бағдарламалық қабат құрылғыларынан сұранысты қабылдайды және оны қалай орындайтынын шешеді. Қарапайым сұраныс мәліметтердің блоктарын оқу болып табылады. Егер ол басқа сұранысқа қызмет көрсетумен айналысып жатқан болса, онда қайта түскен сұраныстардың кезегіне қосылады және ол өзінің кезегі келген кезде орындалады. Енгізу шығарудың сұранысын жүзеге асырудағы бірінші қадам, мысалы, диск үшін, оның абстрактілі формадан нақты пішінге айналуынан тұрады. Дискілі драйвер үшін бұл блоктардың нөмірлерінің цилиндрлердің, бүршіктердің, секторлардың нөмірлеріне айналуын, тексеруді, мотор жұмысы істей ме, бүршік қажетті цилиндрдің басында жатыр ма дегенді білдіреді. Файлды жасағанда немесе оны жаңа мәліметтермен толтырғанда оған жаңа блоктар бөлу керек. Ол үшін ОЖ дисктің бос блоктарының тізімін жүргізуі тиіс. Дискте бос орын бар деген ақпараттық негізінде құрылғыдан тәуелсіз және драйверлер қабатынан жоғары орналасқан бағдарламалық қабатпен жүзеге асырылатын

бос блокты іздеу алгоритмі өңделуі мүмкін.

Қабығы

Пайдаланушы үшін интерфейс жиі нақты пайдаланушының жекелігі. Қабығы қол жеткізуін өзегіне қамтамасыз етеді. Мысалы, GUI қабығы қозғалыс тышқан-қатерлерге ядро рәсімінде болады (мысалы, файлды ашу, қосу, жылжыту файл). Linux және Unix бар қабықшалардағы, командалық жол, сондай-ақ, бұл редакциялау командаларының жолдарын (мысалы, пайдалана отырып, басқару + e өту үшін соңына қарай жолдар, немесе Control + B сақтық көшірмесін жасау үшін бір позицияға) қамтуы мүмкін. Кең танымал қабық GUI Linux / Unix CDE, GNOME және KDE, сондай-ақ танымал мәтіндік қабықта Linux / Unix, Баш (Bourne-Again Shell), Ksh (қабығының Korn) және C қабығының (CSH) кіреді. Мәтінде негізінде қабықты қамтиды, командалық жолдың, тарихын, айнымалылар ортаны және лақап атты меңзейді. Біз Bash қабықшасын 9-тарауда қарастырамыз.

Утилиттер

Бағдарламалық қамтамасыз ету, басқаруға және жінішке параметрлері аппараттық құралдар, операциялық жүйе және қолданбалы бағдарламалық қамтамасыз етуге көмектеседі. Утилиттер қатты өзгеріп тұрады, функцияларды утилиттер файлы (мысалы, Windows, бағдарлама Сілтеуіш терезе), вирусқа қарсы бағдарламалық қамтамасыз ету, қысу файл / дискіні дефрагменттеуді қамтуы мүмкін.

Виртуалды машиналар

Виртуалды машиналарды басқарудың барлық мәселелерін ВМ қосымшасының құрамындағы арнаулы модуль - виртуалды машинаның монитору (ВММ) атқарады. Монитор виртуалды машиналар және базалық қондырғы арасындағы барлық қарым – қатынастарда делдалдық роль атқарады, сонымен бірге бірыңғай аппараттық платформада жасалған барлық ВМ орындалуын қолдап, олардың сенімді оқшаулануын қамтамасыз етеді.

Пайдаланушының виртуалды машинаның мониторуна тікелей қатынауға мүмкіндігі болмайды. Бағдарламалық өнімдердің көбісінде пайдаланушыға виртуалды машинаны жасауға және үйлестіруге арналған гра-

фикалық интерфейс қана ұсынылады. Бұл интерфейс әдетте виртуалды машиналардың консолі деп аталады. Дербес жағдайда, виртуалды компьютер - Linux ОЖ–де іске қосылған арнайы бағдарлама, және ол Intel x86 процессоры негізінде нақты физикалық процессордың моделі десек болады. Виртуалды компьютердің монитормы болып ақпарат шығарылатын Windows XP (Windows NT 4.0 немесе басқа ОЖ) графикалық қоршамының терезесі алынады және виртуалды компьютер келесі виртуалды құрылғылар жиынынан тұрады:

- виртуалды IDE және SCSI қатты дискілері;
- виртуалды CD - ROM;
- иілгіш диск үшін стандартты дискжетек;
- екі бірінші (primary) және екі екінші (secondary) IDE дискілерін қолдайтын 82371 PCI Bus Master - IDE қатты диск контроллері;
- BusLogic BT-958 – мен үйлесімді SCSI – дискілері адаптері;
- стандартты PCI графикалық адаптері;
- стандартты 101/102 – пернелік пернетақта, PS/2 – сәйкес тышқан;
- AMD PCNET Family Ethernet adapter (PCI-ISA) желілік карта;
- COM1-COM4 тізбекті порттары;
- LPT1-LPT2 параллельді порттар;
- Sound Blaster 16 – мен сәйкес келетін дыбыстық карта.

Пайдаланушылық интерфейс

Пайдаланушының жүйемен әрекеттестігін жасақтайтын программалық құралдар кешені; пайдаланушы мен жүйе арасындағы байланыс құралы. Мұндай стандарттар программа қызметтерімен операцияларын қолайлы түрде пайдалануды қамтамасыз ететін экран пішімі мен орындалатын командалардың меню құрылымын тағайындайды. Пайдаланушыға арналған компьютермен (программамен) жылдам әрекеттесу тәсілдері, оның мынадай ерекшеліктерін атауға болады: жүйелік объектілерді (файлдар, құрылғылар, терезелер), меню және мәліметтерді бейнелеу үшін пиктограммаларды (белгішелер) пайдалану; объектілерді тандау, белгілі бір іс-әрекетті бастау, терезені ашу және объектілерді жылжыту үшін сілтеуіш құрылғыларды, мысалы, тышқанды пайдалану; экранды жұмыс үстелі немесе басқару тақтасы ретінде пайдалану. GUI әлдеқайда қарапайым тәсіл, басқару үшін компьютер, ол арқылы берілген команданың көмегімен командалық жолдарды шығарады. Командалық жолы көптеген қапталған Осс * жұмыс істейді. Қабығы бір бөлігі ғана болып табылады, операциялық жүйе, ол ақпаратты енгізуге, оның көмегімен қолданушы басқаруға мүмкіндік беретін ядро операциялық жүйесін бе-

реді. Қабық құрамында мәтіндік ойлап, ол өз командаларының өзегінде операциялық жүйесіне кіреді. Мысал ретінде, қолданушы командалық типті Linux енгізуі мүмкін:

табу~ -name '*' ядро ехес гт {} \;

Бұл команда Linux командасын табу үшін орындайды, осы пайдаланушының үй бағытындағы теориясында, ядро атауымен басталатын, содан кейін кез келген табылған файлда гткомандасын орындайды.

Басқа сөзбен айтқанда, пайдаланушының үй каталогында табылған «ядро» әріпінен басталатын бұл команда барлық файлдарды тауып, жояды. Атап айтқанда, нұсқаулық келесі түрде жұмыс істейді:

1. Издестіру командасы, ол бірнеше параметрлерді анықтау үшін, қабылдайды.
2. ~ / көрсетеді, іздеген жерде көрінуі керек (үй каталогы пайдаланушы).
3. -name 'өзегі *' болып табылады жұп, көрсете отырып, бұл файл аты сұралатын болады әріптер "ядрасы", (The * болып табылады қойылған белгі).
4. -ехес кез келген табылған файл екенін нұсқаулық көрсетеді, ол келесі түрде орындалады.
5. гт операция жою, {} табылған файлды көрсетеді.
6. \;, нұсқаулықты аяқтайды.

1. Бұл ОЖ-гі көпше сандар дегеніміз не? Бірнеше нұсқалары бар, мен ОЖ ыңғайлы болғандықтан таңдадым, бірақ кейбір адамдар мұны жақсы көрмейді, өйткені ОСС мынаны жақтайды: Open Source Software. Басқа адамдар Ож таңдайды. Талқылау үшін <http://weblogs.mozillazine.org/gerv/archives/007925.html>. қараңыз.

Көптеген аудармашылар командалық (мысалы, DOS, қол жетімді Windows) әлдеқайда оңайырақ деп санайды. Linux тартуға қолайлы кейбір пайдаланушылар компьютерлер үшін тәсілді білдіруге өте күрделі нұсқаулықтарды пайдаланады. Сіз Bash 9-тарауда көп жұмыс туралы қалай ойлап, командалық жолдарды қалай шығаратын туралы білесіз.

Процесін басқару

Негізгі міндеті-кез-келген компьютерді іске қосу болып табылады. Компьютерде бағдарлама орындалады ол процесс деп аталады. Себебі ажырата бағдарламасын және процесс болып табылады, онда бағдарлама статикалық субъектісі ретінде процесс болып табылады белсенді объектісін білдіреді. Ликлиникалық процесс бар. Оның мәртебесін қамтуы

мүмкін "жұмыс істейді" (процесс қазіргі процессор уақытта орындалады), "күту" (процесс күтеді енгізу немесе шығару немесе күтіп болу үшін жад), немесе "дайын" (үдеріс жүктеледі жад, бірақ қазіргі уақытта орындалады процессор). Бұл процесс, сондай-ақ бар ерекше сақталатын деректер жадын, кэш-жады мен тіркелімдерді сақтайды. Бұл деректер уақыт және орындау бойынша өзгереді. Осылайша, бұл процесс динамикалық болып табылады.

Компьютер жұмыс істей алады бір процесс болса, немесе бірнеше процестерді бір немесе өзге де нысанда кесіп өту (параллель) сән. Операциялық жүйе жауап береді бастапқы процесін бақылай отырып, ол орындайды, өңдеуді бөлу жағдайлардың (түсіндіруге төмен) және енгізу / шығару (I/O) операцияларды өңдеу көптеген өзара іс-қимыл процестерді және аяқтау процестерді.

Ұзу болып табылады жағдай, қашан процессор үзіледі, оның таңдау-орындау циклі. Айтылғандай 2-тарауында, орталық процессор үздіксіз оқиды, кодтайды орындайды нұсқаулықтар, деректер жад. Сол жақтан өзіне, процессор болады, бұл шексіздікке дейін. Дегенмен, бұған назар процессор тиіс ығысқан ағымдағы процесінің басқа процесіне (операциялық жүйені қоса отырып) немесе жабдықтың бір бөлігі. Осылайша, соңында таңдау-орындау цикл, процессор талдайды туы ұзу (жай тіркелімін туларды), көру үшін, егер кімде-кім көтерді ұзу. Сигнал ұзу мүмкін шығуы аппараттық құралдардың көмегімен немесе бағдарлама. Егер туындайды ұзу, өндейді процессор ұзу былайша. Біріншіден, бұл үнемдейді. Бұл үшін қабылдауға маңызы бар әр түрлі тіркелімдер және сақтауға олардың жадында. Екіншіден, процессор типін анықтайды ұзу көтерілген. Ол үшін процессор талап етіледі анықтау үшін қандай құрылғы көтерді ұзу, немесе, егер ұзу байланысты туындады бағдарламалық қамтамасыз ету. Үшіншіден, процессор ауысады орындау ОС. Атап айтқанда, орталық процессор бастайды орындалуын үзуге өңдеуші. Операциялық жүйе болады өңдеуші ұзу (жиынтық код) әрбір түрі үшін ұзу. Көру

"4.1-кесте ұсынылған тізімі типті үзілісті. Содан кейін CPU орындауды қайта бастайды таңдау-кезең, бірақ енді орындайды өңдеуші ұзу ОС. Орындау аяқталғаннан кейін өңдеуші - Рю, қамқорлық түсірген ажыратылатын жағдайды, процессор түзейді маңызы бар тіркелімдер жадында сақталған. Ол "сергек" үзілген процесс. Содан кейін процессор орындауды қайта бастайды таңдау-цикл, бірақ қазір ол жалғастыруда процесімен ол бұрын жазылған орындаушы, егер ол ешқашан тоқтатқан жоқ. Ұзу ғана қабылдануы мүмкін бірнеше цикл машиналар немесе секундтан бірнеше минутқа дейін, егер ұзу тартуда пайдаланушы. Мыса-

лы, егер ұзу байланысты туындады "диск емес, алып келу", онда қолданушы тиіс физикалық дискіні. Ұзу шешілетін болады ғана кейін ана, қолданушы ретінде түсті.

OSs жүзеге асыру процесін басқару бойынша әр түрлі. Ең қарапайым тәсіл пайдалану бір көп берілген, онда басталады процесс орындалады, және аяқталады қандай да бір басқа да процестер.

Кесте 4.1 Ұзулер типтері

Себебі	Құрылғы (лар) үшін ұзу
Диск	Файл табылған жоқ Диск дисководта жоқ Диск форматталмаған
Пернетақта	Қолданушы Ctrl + Alt + Del пернелерін басады
Тінтуір	тінтуірді басу немесе түсіру
Модем/желі	Хабарлама келеді
Принтер	Басып шығару аяқталды. Бастырғанда қағаздың тұрып қалуы. Қағаздың болмауы
Бағдарламалар	Уақыт қателігі (мысалы, 0-ге бөлу, кіру жаман қолданушы) басқа бағдарламамен байланыс талап етіледі
Таймер	Уақыттың өтуі

Бұл өте қанағаттанарлық пайдалану компьютерлік ресурстар, алайда, сондай-ақ, пайдаланушылар, әдетте, келеді шектеледі қолданумен бір бағдарлама, сол уақытта. Түрлі нысандары параллель өңдеу бар. Параллельді білдіреді процестер орындалады кейбір бөгеттелген fash - ион. Оларға мыналар жатады мультибағдарламау, көп мағыналы (немесе бірлесіп пайдалануға, ол бастапқыда уақыт), және көп ағынды көп процессті . Бұл олардың әрқайсысы біріктіреді, онда операциялық жүйе бірнеше процестер болуы мүмкін, кейбір орындалу жай-күйі. Қоспағанда көп процессті өңдеу (ол пайдаланады бірнеше процессорлар), тек бір процессор, сондықтан бір ғана іс жүргізу мүмкін орындау кез келген уақытта уақыт. Қалған күтеді. Ретінде ұзақ олар күтіп, олар неге күтеді, күтеді ерекшеленеді белгілі бір дәрежеде түріне байланысты басқару процестері.

Біз зерттеп, түрлі нысандары процестерін басқарудың келесі бөлімде. Алайда, жалғастырмас бұрын, біз алдымен енгізілсін көпмәселелік-бұл аталатын бұдан әрі осы бөлімде. "Көп маңызды бар екі немесе одан көп процестердің іске қосылған. Процессор орындайды, тек бір процесс кез келген сәтінде, бірақ өшеді барлық асқынған процестерді тез. Орталық есептеуіш бөлім орындай алады бірнеше мың цикл машиналар әрбір процесс бойынша-0, содан кейін процесс-1, содан кейін 2 процесс, содан кейін қайтара өңдеуге 0. Таймер үшін пайдаланылады уақыт, канша уақыт орындалады, әрбір процесі. Кезде таймер жұмысқа алмаса, ол үзіп процессор және ОЖ, онда шақырылады мәжбүр арасында ауысу процестері. Таймер төгіледі және процессор ауысады келесі үдеріске.

Жоспарлау

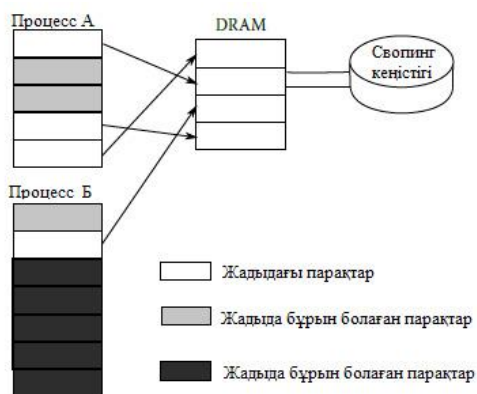
Егер біз компьютерге ештеңе істеу басқа бір өңдеу, яғни, іске қосу-бір бағдарлама, ол аяқталады, содан кейін келесі міндетте пайдаланушының операциялық жүйе орындай жоспарлау. Бірнеше түрі бар кестелер. Мысалы, көп мағыналы режимінде кезде, таймер аяқталады, орталық процессор ауысады ағымдағы процесінің басқа. Ол? Әдетте, операциялық жүйе пайдаланады циклдік жоспарлаушы. Егер бірнеше процестер, жұмыс ерілген, олар орналастырылуы қажет. Содан кейін процестері келтірілген назар БҰ тәртібі, олар кезек. Кезде процессор орындады біраз саны жөніндегі нұсқаулықтарды соңғы үдерісіне кезекте қалпына келтіріледі бірінші процесін кезек. Яғни, кезек-кезек "орау айналасында", осылайша, термин айналмалы. Операциялық жүйе, сондай-ақ жауап береді анықтау, қандай процестер болуы тиіс жүктелген жад кез келген уақытта. Тек процестер, қатысушылар жадында болады кезегіне қойылатын болды. Әдетте, бүгін, барлық процестер жүктелетін болады талап етуі бойынша, бірақ төмен басымдық процестер жойылуы мүмкін кезектегі және дайын жад болуы кезек күту. Және, ақырында, пайдаланушы көрсете алады, бұл процесс іске қосылуы тиіс. Бұл жасалуы мүмкін тандау арқылы, мысалы, процесс қойындысының төменгі бөлігінде жұмыс үстелінің. Бұл мәжбүрлейді көшу үдерісі фон және алдыңғы жоспар. Біз қарастыратын боламыз жоспарлау неғұрлым егжей-тегжейлі келесі бөлімде, сондай-ақ 11-тарауда.

Жадты басқару

Бұрынғы операциялық жүйелерде жадына басқару жай ғана бағдарламаларды және оның берілгендерін (деректерін) кейбір ішкі жинағыштардан (перфоленталар, магнитті ленталар немесе магниттік дискілер) жүктеу арқылы жүргізілді. Мультибағдарламалаудың пайда болуымен

операциялық жүйелерге бар жадыны бір уақытта орындалатын бірнеше бағдарламалардың арасында бөліп беруге байланысты жаңа мәселелер қойылды. Үрдістердің құрылуы кезінде оларға алғаш жадыны бөліп берумен қатар операциялық жүйелер жадыны динамикалық бөлумен де айналысуы керек. Яғни үрдістің орындалуы барысында қосымша уақыт сұранысын қанағаттандыруы тиіс. Қосымша жадыға деген мұқтаждық өтелген жағдайда ол жүйеге қайтарылады. Кездейсоқ ұзындықтағы жадыны уақыттың кездейсоқ аралығында жалпы жадыдан бөлу оның фрагментациялануына алып келеді және осының салдарынан жадыны пайдалану тиімділігі төмендейді. Жадыны дефрагментациялау да жадының бір функциясы болып табылады.

Операциялық жүйеге жұмыс істеу барысында жиі үрдістер мен ағындар сипаттаушылары, ресурстарды бөліп берудің түрлі кестелері, деректер алмасуын болдыратын үрдістерге қолданылатын буферлер, синхронизациялаушы объектілер және т.б. жаңа қызметтік ақпараттық құрылымдар құруға тура келеді. Осы жүйелік объектілердің барлығы жадыны талап етеді. Кейбір операциялық жүйелерде (орналастыру барысында) алдын ала жадының кейбір белгіленген көлемі жүйелік қажеттіліктер үшін қолданылатын қор ретінде сақталады (резервируется). Ал басқа операциялық жүйелерде жүйелік мақсаттар үшін жадыны динамикалық түрде бөлу сияқты қолайлы тәсілдер қолданылады. Мұндай жағдайда операциялық жүйенің әр түрлі жүйеасты жүйелері өздерінің кестелерін, объектілерін, құрылымдарын және т.б. құру барысында жадыны басқарушы жүйеасты жүйесіне сұраныс салады. 4.1-сурет көрсетеді, виртуалды жады екі процестерді (А және Б) әрқайсысы қазіргі уақытта беттерде жүктелетін негізгі жады, ал қалған процестер кеңістікте сақталады.



Сурет 4. Виртуалды жады
Беттер жадында
Беттің бұрынғы жады (шығарылды)

Беттер (әзірге) жадында 101

қойылмаған беттер сілтейді беттер, әлі жұмыс істеуде өтті. Алынған беттер, беттердің, өз кезінде еске, бірақ жойылған болуы үшін мінажат ете жаңа беттері.

Кезде процессор шығарады мекен-жайы жад (мысалы, алу үшін нұсқаулықтар немесе тірек нүктесі), бұл мекен-жайы аударылуы тиіс, оның логикалық немесе виртуалды) мекен-жайы мекен-тұрақты. Орындау үшін түрлендіру, операциялық жүйе қолдайды кестені беттер үшін әрбір процесі. Бұл кесте үшін нені білдіреді әрбір беттің процесс, ол уақытта жады және егер " иә " болса, онда. Сурет 4.2 көрсеткендей, кестелер беттерді көрінетін болады екі процестерді суретте көрсетілген 4.1. Процесі А үш беттер, жүктелген мәңгі сол орындарында, 0, 1, 3 (бұл жерлер деп аталады кадрлармен) және процесс бір бетті жүктелген еске орнында 2. Рұқсат етілетін бағана бит көрсететін болса, бет орналасқан қазіргі уақытта, жад (жарамды бит орнатылған 1) немесе жоқ.

Егер сілтеме бұл бет қазіргі уақытта, жад, кате беттің құрастырып туғызады үзу. Операциялық жүйе қызығады қазіргі уақытта ауыстыруды орындау үшін. Біріншіден, операциялық жүйе табады қолжетімді қадыр пайдалану үшін жад. Егер қол жетімді кадрлар, онда операциялық жүйе таңдау бетті лақтыру. Егер бет өзгертілген болса, онда ол алдымен сақтауы керек кері ауыстыру үшін кеңістік. Кейін кадр қол жетімді болды, ОС жүктейді жаңа бетін кеңістіктен суды алдын-ала айдау бұл кадрда. ОС өзгертеді кесте-беттерді көрсету үшін орналасқан жері жаңа беттен (және, егер бет жадтан жойылған). Және, ақыр соңында, операциялық жүйе процессорге жанарту ағымдағы процесі.

Өйткені кеңістік суды алдын-ала айдау сақталады қатты дискідегі кез келген суды алдын-ала айдау (алмасу) қол жеткізуді қамтитын қатты дискіге. Өйткені уақытта қатты диск жауап әлдеқайда баяу қарағанда, жауап беру уақыты, жад, кез келген орнын ауыстыру баяулауына алып келеді бағдарламасын орындау. Осылайша, үстемелеу аулақ болу керек, мүмкіндігінше. Егер лақтырылады бет болуы қажет жазылған кері ауыстыру үшін орын біріншіден, бұл арттырады уақытта суды алдын-ала айдау.

Процесстің кестелік беттері

Процесс А

Бет	Фрейм	Ұйғарынды
0	1	T
1	–	F
2	–	F
3	3	T
4	0	T

Процесс Б

Бет	Фрейм	Ұйғарынды
0	–	F
1	2	T
2	–	F
3	–	F
4	–	F
5	–	F
6	–	F

Сурет 4.2. 4.1 суретте көрсетілген процестерге арналған беттер кестесінің үлгісі.

Басқа фактор талап ететін жадыны басқару болып табылады, бұл процестер деректер кіру үшін мекен-жайларды туындатады (тиеу және сақтау). Бұл болдырмау үшін, процесс генерациялау мекен-секция жад, ол деректерді қамтиды осы процестің? Жағдай, қашан процесі шығарады мекен-жайы басқа процесін кеңістігін жад деп аталады естің бұзылуы. Бұл болуы мүмкін пайдалану есебінен, көрсеткіштерді, мысалы, бағдарламалау сияқты C және C ++. Естің бұзылуы әкелуі тиіс процесін тоқтату, бірақ C ++, бағдарламашы, сондай-ақ, бұл жағдайды шешу арқылы өңдеушінің ерекшеліктерді білдіретін өңдеуші ұзу жазылған бағдарламашы енгізілген және ағымдағы бағдарламаны.

Ресурстарын басқару

Басқа жад және процессор, көптеген басқа да ресурстар, қол жетімді, оның ішінде файлдық жүйесінің желісіне, сондай-ақ басқа құрылғыларды пайдалану. Операциялық жүйе қолдайды кестені барлық белсенді процестер мен ресурстар, бұл әрбір процесс қазіргі уақытта пайдаланады немесе пайдалануға қалайды. Көптеген ресурстар қол жетімді болуы мүмкін тек в бірін-бірі жоққа шығару. Яғни, процесін бастайды пайдалануға, ресурс, басқа ешқандай процесін пайдалана алмайды ресурсы, әзірге бірінші процесс освободит. Босатылғаннан кейін, операциялық жүйе шеше алады, қандай процесс келесі қол жеткізуге мүмкіндік алады ресурсы.

Көптеген себептер бар, өзара болдырмау, бірақ қарап, осы жағдайды. Файлды сақтап, ағымдағы шот теңгерімі. Айталық, қазіргі уақытта \$ 1000. Енді елестетіңіз, сіз және сіздің елеулі басқа кіру-ақ, кезекте, сізге көмектеседі әр түрлі кассирлері бір мезгілде. Сіз де сұрайды шегеруге \$ 1000. Сол уақытта, екі кассирлер алуға деректерге қол жеткізу және табу \$ 1000 қолжетімді. Мұнды, сіз шотты жабу, олар екеуі де енгізуге мәміле, қайтадан сол уақытта. Екі компьютерлерде tell - ERS "қолжетімділік сол жалпы файл, диск, бір, сақтайтын ақпаратты есепке алу жазбасы туралы. Бір мезгілде екі оқу бағдарламасына маңызы бар (\$ 1000), анықтау, ол көп немесе оған тең, сіз кері қайтарып алады, және алып тастау мәні \$ 0. Сіз және сіздің маңызды басқа әрбір серуен \$ 1000, ал қалдығы дейін қысқартылды \$ 0. Қарамастан, онда бар, тек \$ 1000 бастау үшін, енді сізде \$ 2000! Бұл жақсы жаңалық, сіз үшін, бірақ өте жаман жаңалық. Мұндай жағдайды болдырмас үшін болған жоқ, біз бір-келкі емес қол жеткізу кез-келген жалпы ресурсқа ол жеткізу кез-келген жалпы тіректік нүктесіне тиіс синхронды болып табылады нүктесі байланыстыру арқылы бірігіп компьютерлік желі немесе арасында бөлінеді екі процестермен жұмыс істейтін бір уақытта бір компьютерде. Егер бір процесс бастайды қол жеткізу тірек нүктелері, басқа ешқандай процесс емес тиіс рұқсат оған әзірге бірінші процесс аяқтайды қол жеткізу және қорды босатады. Операциялық жүйе өндеуге процесс ортасында синхрондауды қамтамасыз ету үшін қарама-қарсы ресурстарға қол жеткізу. Десекте біздің мысалға есепші банктің қадамдастыру бірлесіп пайдаланатын есептік жазбасын тексеру нөлдік нүктелері жұмыс істейтін болады, ол. Операциялық жүйе болады, екі сұрау үшін қол жеткізу ортақ файлға. Операциялық жүйе бір, екі процестер үшін рұқсат беру. Басқа да күтуге мәжбүр болады. Осылайша, онда бір есепші орындай алады операциясын деректер базасын шегеруге \$ 1000, ал екіншісі тура келеді

күте тұру керек. Кейін бірінші кассир жасалады, қор босатылады, ал екіншісі кассир енді қол жеткізу мүмкін файл. Өкінішке орай, екінші кассир табады балансын құрайды \$ 0, \$ 1000 емес, жұртқа ешқандай ақша. Осылайша, сіз және сіздің екінші жарты тек қана кетіп \$ 1000. Басқа tracking ашық ресурстардың, операциялық жүйе, сондай-ақ өңдейді. Қарапайым нұсқасы тығырыққа тірелген 4.3-кестеде көрсетілген. Операциялық жүйе көп мағыналы арасындағы процестерді P0 және P1, және бар (кем дегенде) екі қолда бар ресурстар, R0 және R1.

P0 жұмыс істейді біраз уақыт, содан кейін қол жеткізуді сұраған R0.

Операциялық жүйе екендігін анықтайды басқа ешқандай процесс пайдаланады R0, сондықтан ұсынады сұраныс.

P0 жалғасуда, әзірге таймер емес, жұмысқа алмаса, операциялық жүйе ауысады P1.

P1 жұмыс істейді біраз уақыт, содан кейін қол жеткізуді сұраған R1.

Операциялық жүйе анықтайды, бірде-бір басқа процесс емес, пайдаланады R1, сондықтан рұқсат беруге сұрау салу.

P1 жалғасуда, әзірге таймер емес жұмысқа алмаса және ОС ауысады P0.

P0 жалғастыруда пайдалануға R0, бірақ енді, сондай-ақ сұрайды R1.

Операциялық жүйе екендігін анықтайды R1 білдіреді қазіргі уақытта пайдаланылады P1 және ауыстырады P0 кезекте күту ауысады P1.

P1 жалғастыруда пайдалануға R1, бірақ енді, сондай-ақ сұрайды R0.

Операциялық жүйе екендігін анықтайды R0 қазіргі уақытта пайдаланылады P0 және ауыстырады P1 кезекте күту.

Операциялық жүйе орындауға дайын процесін, бірақ және іске қосылған процестер, P0 және P1, тұр күтіп, кезек. Сонымен қатар, бірде-Р0, бірде-Р1 бастауы мүмкін емес, қайтадан, әзірге ресурс, оны күтеді (R1 және R0, тиісінше) қолжетімді болады. Бірақ әрбір процесс ұстайды, интернет, басқа да қажеттілігін, және әрбір күтеді, ешқашан уақыт ресурс қол жетімді болу үшін не процесін бастау үшін қайтадан,

осылайша, ең соңғысы.

Күресу тұйығы үшін, кейбір тексеру операциялық жүйелерді көру үшін, егер өзара бітеу туындауы мүмкін ұсынар алдында қандай да бір сұранысы. Бұл, әдетте, сақтандырылады, көпшілік операциялық жүйелер. Басқа операциялық жүйелер жұмсауға біраз уақыт сайын бір рет біраз уақыт көру үшін, егер тұйық туындады. Егер иә болса, онда бір немесе бірнеше кіргенде тұйық процестер еркін өлтірілді, босату сезініп, өз ресурстары және мүмкіндік беретін басқа процесс (лар) жалғастыру үшін. Процестер қандай да бір кездейсоқ уақытта қараусыз қалған



Сурет 4.3. Тығырықты жағдай

Тағы басқа да операциялық жүйелер ісі болуы мүмкін тұйығы және барлық. Бұл пайдаланушыға дейін шешу егер тұйық туындады және тиісті шаралар қабылдауға!

Басқарудың файлдық жүйесі

Негізгі мақсат-ерте USS ұсынуға пайдаланушыларға қол жеткізу мүмкіндігі және басқару файлдық жүйесімен (аты MS-DOS білдіреді Microsoft Disk Operating System, ал командалар толықтай дерлік болды іс файлдық жүйенің дискідегі). Типтік командалар файлда басқарушы тиіс файлды ашу, файлды жылжыту, атын өзгерту, файлды жою файл басып файл, сондай-ақ жасауға, жылжытуға, атын өзгертіп және жою каталогтар (папкалар). Бүгін осы мүмкіндіктерді табуға болады файлда бағдарламалар адам-Ager сияқты Windows Explorer. Әйтсе де, бұл команданың орындалуы мүмкін арқылы тасу басу арқылы, және т. б., олар сондай-ақ қол жетімді пәрмен жолынан. "Unix / Linux, файлдық командалар жүйесін қамтиды компакт-диск, LS, мв, APC, P. M., MkDir, Rmdir және Ыр. DOS, команда файлдық жүйесін қамтиды компакт-диск, реж,

жылжытуға, көшіруге, дель, MkDir, Rmdir және мөр. Бар кең қосымша командалар Unix / Linux мүмкіндік беретін жүйелік әкімшісіне басқару файлдық жүйемен тұтастай алғанда орнату арқылы жаңа құрылғыларды немесе жылжыту, онда бұл құрылғылар қол жетімді файлды кеңістікте, сондай-ақ тұтастығын тексеру файлдар (мысалы, кейін, файлды ашық болып қалады, және жүйе өзі қайта жүктеу).

Сахна артында, операциялық жүйе басқарады қол жеткізуді файлдарға. Пайдаланушы жолдайды командасын осындай жылжыту үшін файл немесе файлды ашу, ол тікелей пәрмен жолынан немесе кейбір бағдарламалық қамтамасыз ету, пайдаланушы шындыққа ұсынған сұрау. Операциялық жүйесі қазіргі уақытта өзіне алады. Біріншіден, операциялық жүйе кепілдік беруі тиіс, бұл қолданушы құқығы қол жеткізу үшін сұрау салынатын файл.

Екіншіден, операциялық жүйе файлды тауып. Дегенмен, пайдаланушылар орналасқан жерін көрсету файлдарды, бұл жерлер болып табылады логи - орналасуы кал файл файлдық жүйесі. Мұндай ерекшелік камтиды бөлім диск және каталогтық файл (мысалы, C: \ Users \ Foxr \ Менің құжаттарым \ CIT 130). Алайда, талқыланатын болады 5-тарауында, файлдар сынған неғұрлым ұсақ бірліктер. Операциялық жүйе анықтау керек файлдың бір бөлігін мүмкіндігінше, және оның физикалық орналасқан жері. Операциялық жүйе көрсетуі логикалық орналасуы да жеке орналасқан жері, ол тұрады бетінің диск орны бетіне. Сайып келгенде, операциялық жүйе болуы тиіс бастамашылық байланыс дискімен орындау үшін қажетті операция.

Қорғау және қауіпсіздік

Көптеген компьютерлер бұл күндері бір бөлігі болып табылады үлкенірек желілер компьютерлер, мүмкін бірге пайдаланылатын файлдар, желісіне, немесе қандай да бір басқа бірлесіп пайдаланылатын ресурс. Сонымен қатар, бірнеше пайдаланушылар бірлесіп пайдалануға бір және сол компьютер. Файлдар сақталатын ретінде жергілікті, сондай-ақ желісі арқылы қорғалуы тиіс болатындай қолданушы кездейсоқ емес (немесе арам оймен) қайта жазу, өзгертуге немесе дұрыс емес файлдар басқа пайдаланушы. Қорғау, кепілдік береді, бұл пайдаланушы зиян келтіру жүйелік емес пайдалана отырып, бөтен файлдар, жүйелік ресурстар зиян келтіру және т.б. қауіпсіздігін қорғауды қамтамасыз етеді. Жалпы механизмі қорғау және қауіпсіздігін қамтамасыз ету болып табылады. Әрбір қолданушы болады пайдаланушының аты және сәйкестендіру; механизмі (көбінесе пароль). Кейін операциялық жүйе орнатты,

кім болып табылады, қолданушы, қолданушы содан кейін қол жеткізу мүмкін ресурстар мен файлдарға, пайдаланушы құқылы қол жеткізу. Олар қамтитын болады жалпы файлдар (файлдар, күні кейде басқа да пайдаланушылар үшін қол жетімді) және өздерінің жеке файлдар. Unix және Linux пайдаланады өте тәсіл орналастыру арқылы әрбір пайдаланушы тобына. Файлдарды оқуға, жазуға және өлім жазасына кесілді, сондай-ақ қол жетімділігі түріндегі файлды басқаруға болады, қамтамасыз ету үшін әр түрлі құқық иесінің файл, топ файл мен қалған әлем. Мысалы, сіз файл жасау мүмкін күнінен бастап дәл хронологиялық жасы / аталған қабілетті / орындалушы файл өз бетімен оқу / орындалатын кім-сіздің топта атқарылатын. Қарастырайық тіркелгілер пайдаланушылардың, топтардың және бақылау әдістері қол жеткізу 6-тарауда.

ҮДЕРІСТІ БАСҚАРУ ТҮРЛЕРІ

Әр түрлі тәсілдері, олардың ПСОК бағдарламаларды орындауға болады.

- Бір мәселелік
- Пакеттік
- Көп бағдарламалық
- Көп мәселелік
- Көп ағымдық
- Көп процессерлік

Бір мәселелік жүйесі, ең ескі және ең қарапайым, жай орындайды, бағдарлама, әзірге ол емес деген қорытындыға, ал содан соң ауысады кері ОС күтіп пайдаланушыға сұрау басқа да бағдарламасын орындау үшін. MS-DOS-тың мұндай ОС. Барлық бағдарламалары орындалды тәртібі, олар сұратқан. Егер сайлау процесін талап етеді, назар сияқты созылған енгізу немесе шығару, процессор болады жүрісте, әзірге пайдаланушы немесе құрылғы енгізу / шығару емес, деп жауап берді. Бұл өте тиімсіз. Ол сондай-ақ қайғылы болуы мүмкін, пайдаланушылар үшін артық бір нәрсе бір рет жасағысы келеді. Басқа ерте компьютерлер, ең ерте мэйнфреймдер, сондай-ақ жеке көп мәселелік.

Жүйесі пакеттік өңдеу ұқсас бір жүйеде қойылған міндеттерді қоспағанда, бар кезек күту желісі) үшін процестер. Басты айырмашылығы осында деп пакеттік өңдеу жүйесі бар бір пайдаланушы. Осылайша, пайдаланушылар ұсынуға өз бағдарламасын орындау үшін кез келген уақытта. Сұрау берген кезде бағдарламасына процесі кезекке қосылады. Кезде процессор аяқтап, бір процесс, ОС шақырылады шешу үшін қандай

күтудегі үрдістерді жеткізу үшін CPU толықтырылсын. Кейбір нысандары жоспарлау шешу қажет, қандай процесс таңдап алады келесі. Жоспарлау алгоритмдері қамтиды сұлбасын басымдықтарын, қысқа жұмысын алдымен, бірінші келді-бірінші қызмет көрсетілді. Үшін сұлбасын басымдық процестердің әрбір аталған басымдық мәртебесіне байланысты пайдаланушының (мысалы, әкімшілік, факультеті, аспиранты, студент). Алгоритм негізінде жоспарлау, жүктеме машинаға, және сіз процесін орналастырылады кезекте, сіз таба күту бірнеше минут, сағат немесе тіпті күн бұрын, сіздің бағдарлама орындайды! Қысқа жұмыс біріншіден, ОС тырысады бағалауға саны процессорлық уақыттың әрбір процесс талап етеді және таңдайды процесс, ол алады ең аз уақыт. Статистикалық, бұл сақтайды күтудің орташа уақыты минимумға дейін, бұл көрінуі мүмкін әділетсіз пайдаланушыларға қатысты, олар процестер көп уақыт алатын. Біріншіден келді-бірінші қызмет көрсетілді сұлбасы болып табылады дәстүрлі, кез келген кезектегі болсын, компьютерде немесе қалай табуға болады? Бұл сұлба белгілі және, сондай-ақ FIFO (бірінші күні) болып табылады әділ, бірақ міндетті емес тиімді.

Басқа арасындағы айырмашылық бірыңғай жүйесімен міндеттерді қою мен жүйелері пакеттік өңдеу деректер болып табылады пакеттік өңдеу жүйесі деректер жоқ интерактивті пайдаланушы. Бір міндеттерді қою, егер I / O талап етіледі, процесс токтатылады, әзірше компьютер күтеді енгізу немесе шығару үшін аяқтау. Бірақ пакеттік деректерді өңдеу, барлық кіру деректері ұсынылуы тиіс, сол кезде қолданушы жібереді. Өйткені пакеттік өңдеу ең көп таралған алғашқы үш ұрпақ компьютерлер, әдетте, кіру енгізілді перфокарталарды, немесе сақталады магнитті таспаға. Сонымен қатар, барлық шығу мәліметтері өңделетін болады "желіден тыс", жіберіледі магнитті таспаға немесе принтер. Әлбетте, интерактивті, көптеген түрлері процестерді жұмыс істейтін болады басқаша, біз үйреніп қалған. Мысалы, компьютерлік ойын талап етеді, барлық жолдары пайдаланушының көрсетілген мерзімге дейін ойындар, мәтіндік процессор талап етеді, барлық редакциялау және өзгерістер пішімдеу көрсетуге іске қосар алдында, мәтіндік процессор. Сондықтан, пакеттік өңдеу, деректерді өз шектеулері бар. Оның орнына, пакеттік өңдеу үшін кеңінен қолданылады секілді міндеттерді есептеу жалақысының немесе салықтарды немесе басқа математикалық есептеулер.

Назар аударыңыз, бұл пакеттік өңдеу, бір міндеттерді қою, тек бір процесс орындалады, сол уақытта, қоса алғанда операциялық жүйесі. Операциялық жүйе шақырылады кезде жаңа процесс сұралды пайдаланушы. Бұл болған қолданушы жаңа сұрау салуды жібереді.

Жеке жүйесі-оқтын жаппай қызмет көрсету үшін жауапты қабылдау

жана түсініктер пайдаланушыларды қосу, оларды тиісті кезек.

Дегенмен пакеттік өңдеу таралды компьютерлерде бірінші-үшінші генерациялау жұмыс істейтін мейнфреймах және мини-компьютерлер, сіз әлі таба аласыз кейбір партия өңдеу бүгін. Мысалы, Unix және Linux бар жоспарлау команда сияқты созылмалы және, сондай-ақ Windows, бағдарлама тапсырмаларды жоспарлаушы қол жетімді болып табылады. Бағдарламаның кесте бойынша жұмыс істейтін болады үздіксіз, олар жұмыс істейді пайдаланушының араласуынсыз, бірақ олар жұмыс істей алады көп мағыналы режимінде қандай.

Жүйесінде пакеттік деректер кіру қол жетімді, қашан процесі басталады орындау арқылы қандай да бір көзі сияқты перфокарталар немесе файл магниттік таспадағы немесе дискідегі. Бір жүйесі қойылған міндеттерді орындау, екінші жағынан, would енгізу тікелей пайдаланушымен пернетақта көмегімен немесе басқа да құрылғылар. Ұсына отырып, пайдаланушының (адамның) процесін баяулатып, өңдеу. Не үшін? Өйткені адам соншалықты әлдеқайда баяу, ақпарат келіп түскен кезде, ол диск таспалы жинақтағыш немесе есептеуші перфокарталарды (тіпті егер оқырман перфокарталар өте баяу). Сонымен қатар, күту шығу баяулатуы мүмкін өңдеуді, тіпті егер шығу өңделеді желіден тыс. Жылдамдығы магниттік таспаның, дискінің немесе принтер, әлдеқайда баяу карағанда, орталық процессор. Содан кейін бірыңғай жүйесі қойылған міндеттерді елеулі тиімсіздігі-кіру және шығу қорғанысын өңдеу. Бұл сондай-ақ қолданылады және пакеттік өңдеу, тіпті егер пакеттік өңдеу емес, айтарлықтай әсер, өйткені бірде-бір адам циклында.

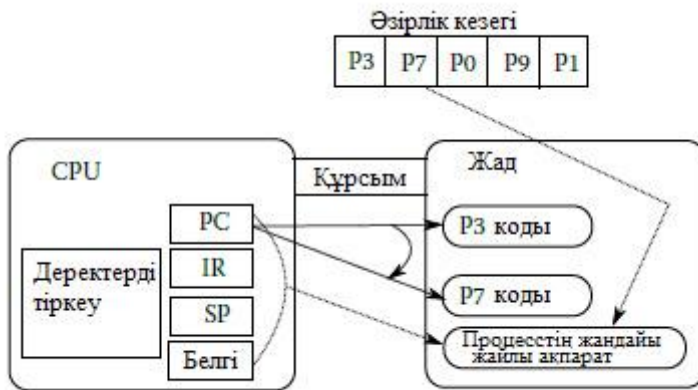
Жүйесі мультибағдарлауға ұксас жүйесі мерзімдік жүктеу қоспағанда, егер ағымдағы іс жүргізу талап енгізу/шығару болса, онда бұл процесс жылжиды басқа кезек (қылтасы кезегін іске қосу/күту), ал операциялық жүйе таңдайды басқа процесін орындау үшін. Кезде бастапқы процесс аяқталады, оны енгізу / шығару, ол жаңартылады және ауыстыру процесі жылжиды кері кезек. Осылайша, ұзақ уақыт енгізу / шығару тудырмайды процессор қалуы күту режимінде, сондықтан жүйесі әлдеқайда тиімді болып табылады. Идея тапсырып процессор операцияларды орындауға енгізу / шығару деп аталады кооперативтік көп мәселелік. Біз осы идеяға төменірек ораламыз.

Бірнеше бар әр түрлі қолдану үшін кезекті көп мәселелік, сондықтан бізге қажет олардың айырмашылығы. Бар дайын кезек (кезек процестердің күтетін үшін CPU), кезек енгізу / шығару (бір-бірден және әрбір құрылғы үшін I / O), және кезек күту (кезек процестердің сұратқан іске қосу үшін, бірақ әлі орналасуға кезекке дайын). Себебі, бар кезекте күту бұл процестер кезегі дайын қазірдің өзінде жұмыс істеуде өтті. Өйткені

жад шектелген мөлшерде болуы мүмкін процестер, сұратылған пайдаланушылар алмайтын сия, сондықтан олар отыр кезек күту жоқ босатылады орын кезекте дайын. Онда бөлме дайын кезек, егер процесс аяқталады, және бұл, кезек немесе егер көптеген процестер болды ауысуы кезек дайын кезек енгізу / шығару.

Жүйесі мультибағдарлау талап етеді қосымша тетігін, ретінде белгілі контексте коммутатор. Ауысу контекст ғана болып табылады CPU бір процестің басқа. Біз зерттедік, бұл мәселені қысқа бұрын. Кел жақынырақ. Ауысу үшін, процестердің, орталық процессор алдымен сақтап, ағымдағы процесс "жай-күйін және алу процесінің келесі мәртебесі". Сурет 4.4 қамтамасыз етеді. Компьютер үшін мәнін ауысуын суреттейді. Бұл кезеңде процесс P3 орындалады. Тізілімінде PC сақтайды, мекен-жайы жад келесі команда P3, ИК ағымдағы нұсқаулық сақтайды P3, СП сақтайды стек шыңына орындау кезінде P3 еске жалаулар сақтау мәртебесі соңғы команда орындалған, мен тіркелімдер деректерді сақтауға тиісті деректер үшін P3. Ауыстырып қосқыш контекст талап етеді бұл мәндер жадында сақталады. Содан кейін, P3 жылжиды тиісті кезек енгізу / шығару жад, орналастыра процесі P7 - майданға дайын кезек. Ақырында, процессор қалпына келтіруі тиіс мәртебесі P7 барғанда, еске түсіріп, сақталған маңызы бар тіркелімдер үшін P7. Енді компьютерге сақтауға болады мекен-жайы жад келесі нұсқаулықты P7, ИК сақтап, ағымдағы нұсқаулықтың P7, ОС сақтайды жоғарғы бөлігінде стек орындау кезінде P7, жалаулар сақтап, мәртебесін соңғы орындалған нұсқаулықтың P7, мен тіркелімдер деректерді сақтауға болады соңғы мәліметтер пайдаланылатын P7. CPU орындауды қайта бастайды таңдау-цикл, бірақ енді P7 емес, P3. Бұл жалғасуда алмайынша не P3 немесе P7 талап етеді енгізу / шығару, ол өз жұмысын аяқтайды. Бірінші жағдайда, ауысу мән мәтін жүреді, онда P3 қалпына келтіріледі және P7 кезекте, ал соңғы жағдайда, процессор ауысады бастап P7 арналған P0.

Өйткені ауысу мән мәтіні талап етеді, сақтау және қалпына келтіру мәтіннен екі процестер, ол лезде жүреді, ал бірнеше үлкен уақыт шығындарын.



4.4-сурет Ауысу контекст өзгерту мәндерінің тіркелімін талап етеді.

Ажыратқыш. Мән мәтін ауысуын талап етеді, бірнеше мүмкін ондаған операцияларының жады бар. Бұл баяу салыстырғанда жылдамдығы процессор. Егер ауысу контекст болуы мүмкін сақтауға және қалпына келтіруге бірі кэш, соғұрлым жақсы. Дегенмен, кейбір жоғары өнімді компьютерлер қосымша жинақтар регистрлерді ауыстырып қосу үшін контекст. Бұл қамтамасыз етеді жылдам үн қосу және, осылайша, сақтайды тұрып қалу уақыты процессор кезінде ең аз уақыт. Дегенмен, қарамастан тұрып қалу уақыты процессор, шақырылған ауыстырып қосу арқылы контекстен көмегімен ауыстырып-қосқыш контекст "көптапсырмалық әлі күнге дейін әлдеқайда тиімді мүмкіндік тұрып процессор операция кезінде енгізу немесе шығару.

көптапсырмалық қабылдайды идеясын көптапсырмалық бір қадам алға. "Multiprogram - мина, процесі ғана тоқтатылды (мәжбүр күтіп), ол қажет операциялар, енгізу / шығару, ең оятатын CPU, ауысу үшін басқа процесс. Бірақ көп мағыналы режимде таймер үшін пайдаланылады санын есептеу тактілері өткен сәттен бастап соңғы ауысу контекст бастады, бұл процесс. Таймер белгіленеді біраз бастауыш маңызы бар (мысалы, 10000). Әрбір жаңа тактовый цикл, таймер азаяды. Кейін ол жетеді 0, таймер тоқтатады процессор мәжбүр оның ауысу басқа процесс. Сурет 4.5 білдіретін нұсқа сурет 4.4 көрсетеді ауысу контекст ретінде мәжбүрлі таймермен. Процесс ілуге алады орналасу соңына кезек дайын болуы тиіс-да. Бұл сияқты көрінуі мүмкін үлкен айыппұл процесін, бірақ шын мәнінде, өйткені қазіргі заманғы процессорлар сондай-ақ тез,

жарамды процесс күтуде, тек бірнеше миллисекундтан, ең көп бұрын, ол жылжиды кері CPU. "Көп мәселелік жүйелерінде, адамдар тіпті байқамайды, тоқтап, содан кейін қалпына келтірілді, өйткені, жауап беру уақыты адам көп миллисекунды. Сондықтан, дегенмен компьютер ретінде ұсынылады, EHE - пішу бірнеше бағдарламаларды бір мезгілде ккп мағыналық, шындық мынада: компьютер орындалды процестер қатарлас, немесе бірін-бірі қайталайтын, сән.

"Көптапсырмалық, біз екі механизм, оның көмегімен процессор көшетін болады және бір процесс екіншісіне: өйткені ағымдағы процесс сұратады кейбір нысанын енгізу / шығару немесе, өйткені таймер жұмысқа алмаса, бұл процесс. Көп мәселелік дұрыс болып саналады кооперативтік көп мәселелік болғанда процесс өз еркімен бас тартып, процессор.

Windows NT операциялық жүйесі басынан бастап жаңа операциялық жүйелерге қойылған талаптарға сай жасалды: кеңейтілуінде, ауыстырылуында, сенімділікте, жаңалығында, өндіруде. Бұл қасиеттер клиент-сервер, микроядро, объектілер сияқты алдыңғы қатарлы құрылымдық бағдарлаудың технологиясын қолданудың арқасында пайда болды. Windows NT-да ығыстырумен (preemptive multitasking) көптапсырмалылық механизмі қолданылады. Windows NT Server жіптерімен басқару үшін приоритеттер механизмі қолданылады. Белгілі бір уақыт мезеттерінде приоритеттерді бағалау мен жіптердің қайта жіктелуі қарастырылады, мұның нәтижесінде бір ағын бағдарламасының кезек деңгейлері әр түрлі процестермен орындала алады немесе келесі процессордың босағанына дейін қала береді.

Windows NT сәйкес операциялық жүйесі кез келген бос процессорда немесе олар арасында жадыны бөліп, барлық процестерде бір мезетте орындала алатын симметриялық көппроцессорлы есептеме процесін қабылдайды. Көптапсырмалылық жіптер деңгейінде орындалатынын ескерсек, бір процессордың әр түрлі бөлімдері қатар орындала алатынын айтуға болады. Бұдан көп- жіпті серверлер бір клиенттен артық жібере алады. Windows NT Server 16 қатар процессорге дейін жұмыс істейді, бұл 16 процессорлі Intel 486/50 МГц Sequent фирмасының Symmetry 750 сияқты серверлеріне ыңғайлы. Windows NT Server-дің СМП өткізілуі өндірістің үйлесімділігіне бағытталған және бұзылуларға қарсы тұру мақсатында қайта сақтауды қарастырмайды. Бір процессордың бұзылу мезетінде жүйе тоқталады.

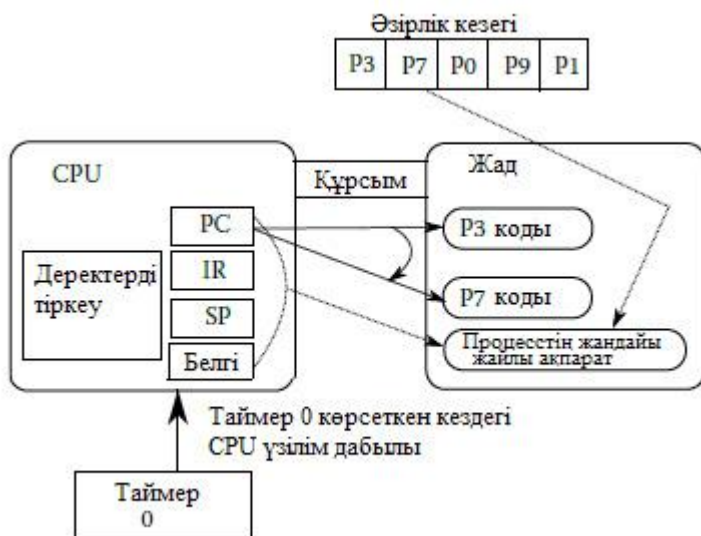
Windows NT Server-де толықтай СМП архитектурасының қарқынды ауқымы потенциалы жасалған. Бірпроцессорлі жүйені ОЖ түрін немесе қосымшаны өзгертпей, процессорлар санын өсіріп, оңай дамытуға болады.

Windows NT Server кіріс/шығыс құрылғыларымен басқаруда асинхронды әдіс қолданылады. Процестің бітуі үшін немесе жаңа тапсырманың басталуы үшін оқу не жазба сияқты операциялардың бітуі жайлы хабарды тосу қажет емес. Әр процесс бағдарламаны спецификалық көрсетумен орындалатын бір жіпті қолданумен жасалады. Соңында бағдарлама жаңа жіптер жасай алады, және Windows NT Server бұған үлкен деңгейлі қосымшаны қолданбай-ақ, оларды жіктейді және олармен басқарады. Қолданбалы бағдарлама бірнеше ағындарды қолдану үшін мұны оның алгоритмінде жасаудың керегі жоқ. Бөлек ағын әр операция үшін жасалады. Мысалы, бір ағында бағдарлама күрделі сызба пішінін жасаса, екіншісінде жөндеу үшін объемды сызбаны қолданады. Бұл ағындардың әрбіреуі қосымша жағынан ешқандай қатысуды талап етпей бөлек процессорларда істейді. Процессор ішіндегі ағындар жадының ортақ облысын қолданады, сәйкесінше олар әдейі деректермен алмаспау керек.

Бірігу талабына сәйкес, Windows NT Win32 API бағдарламалық интерфейсті қосымша үшін ғана орындау ортасын қамтамасыз етпейді. Intel фирмасының процессорларында орындауда Windows NT сақтанған ішкі жүйелері MS-DOS, Win16, OS/2 қосқанда Microsoft мекемесінің бар қосымшаларының екілік сәйкестігін реттейді. MIPS RISC процессорларында екілік сәйкестік MS-DOS қосымшалары және 16-битті Windows-қосымшалары үшін (эмуляцияны қолданумен) жетеді. Windows NT, сонымен қатар, IEEE 1003.1. үлгісінде анықталған интерфейсін қатаң ұстанатын POSIX-қосымшалар үшін бар тестер деңгейінде сәйкестік қамтамасыз етіледі.

Бағдарламалық интерфейс сәйкестігінен басқа Windows NT MS-DOS (FAT), CD-ROM, OS/2 (HPFS) және жаңа (NTFS) файлдық жүйелерін қолдайды.

Көптеген басқа операциялық жүйелерге қарағанда Windows NT басынан бастап желіде жұмыс істеуге арналған. Нәтижесінде файлдарды, құрылғылары мен объектілерді ортақ қолдану функцияларын қолданушы интерфейсіне енгізілген. Басқарулар желі жұмысын орталықтандырылған басқарумен қадағалай алады. Ең маңыздысы клиент-сервер типті қосымшалар көпкомпьютерлік жүйелерде жұмыс істей алады.



Сурет 4.5.Көпмағыналық контексіне ауысу

Тағы бір себебі-процесс мүмкін өз еркімен бас тартуға CPU тиіс не күту сәтке кездесуге. Кездесуге жүреді, бір процесс күтеді біраз ақпаратпен (шығу, жалпы нөлдік нүкте) басқа процесс. Тағы бір себебі процесс үшін тапсыруға CPU болып табылады, бұл процесс төмен басымдық және басқа да, аса жоғары процестер басымдық күтеді.Бастапқы идея көп мәселелікдеп аталды бойынша бөлу. Режимінде уақыттың бөліну процесі ре түспеген процессор дербес. Тек кейінірек, OSs іске асыруға кірісті бәсекелестік көптапсырмалы бірлесіп пайдалану термин уақытта құлады. Уақыттың бөліну алғаш рет іске асырылды мейнфреймах 1957 жылы, бірақ жиі қолданылды үшінші ұрпағына дейін. Төртінші ұрпаққа, бұл әдеттегі құбылыс мейнфреймдер, бірақ жеке компьютерлер дейін 1980-х. шын мәнінде, ДК бастап Windows ОС дейін қатар Windows 95 және Mac OS бұрын OS X орындалды кооперативті, бірақ бәсекелестікке шыдамайды, көптапсырмалық.

Процестер туралы айтқанда, операциялық жүйе олардың жекешенушілігін қолдайтынын ескеруге болады: әрбір процестің өзінің виртуалды адрестік кеңістігі болады, әрбір процеске өзінің қорлары – файлдар, терезелер, семафорлар, т.б. тағайындалады. Мұндай жекешеленушілік бір процесті басқасынан қорғау үшін қажет, себебі олар машинаның

барлық қорларын бірлесіп қолданатындықтан, бір-бірімен бәсекелеседі. Жалпы жағдайда процестер бір компьютерді қолданатын әр түрлі пайдаланушыларға жатады және ОЖ өзіне процестердің қорлар үшін таластарында төреші рөлін атқарады.

Мультибағдарламалауда жүйенің өткізу мүмкіндігі өседі, бірақ бқлек процесс, егер деол бірбағдарламалық тәртіпте орындалатынына қарағанда ешқашан жылдамырақ орындала алмайды. Алайда, бір процестің шеңберінде шешілетін міндет оның шешілуін негізінде тездетуге мүмкіндік беретін ішкі параллелезизмге ие болады. Мысалы, міндеттің шешілу барысында сыртқы құрылғыға жүгіну пайда болады, сондықтан осы амалдың орындалуы кезінде процестің орындалуын толығымен құрсауламай-ақ қойып, процестің басқа «тармағы» арқылы міндеттеуді жалғастыруға болады.

Осы мақсаттар үшін қазіргі ОЖ-лер көптармақты өңдеу(multithreading) механизмін қолданады. Бұл жағдайда «жіп» (thread) деген жаңа ұғым енгізіледі.

Мультибағдарламалау енді жіптер деңгейінде жүзеге асады, сондықтан бір процестің шеңберінде бірнеше жіптер түрінде құрылған есеп оның бүлек бүліктерінің жалған параллельді (немесе мультипроцессорлық жүйеде параллельді) орындалуының арқасында жылдамырақ орындалуы мүмкін. Әсіресе, таратылған қосымшалардың орындалуы үшін көпжіптілікті қолданған нәтижелі, мысалы, көпжіпті сервер қатарлас бірден бірнеше клиенттердің сұраныстарын орындай алады.

Бір процеске қатысты жіптер дәчтүрлі көпмақсатты жүйедегі процестерге қарағанда бір-бірінен соншама оңашаланған емес, олардың арасында тар әрекеттестікті ұйымдастыру жеңіл.

Жіптерді кейде жеңілдетілген процестер немесе мини-процестер деп атайды. Әрбір жіп қатал жүйелі түрде орындалады және оның өзінің меншікті бағдарламалық санауышы мен сүрлеуі болады. Жіптер процестер сияқты, мысалы, жіптер ұрпақтар туғыза алады, қалып-күйден қалып-күйге кешіп өте алады. Жіптер келесі қалып-күйлердің біреуінің орнында бола алады: ОРЫНДАУ, КҮТУ және ДАЯРЛЫҚ. Бір жіп құрсауланғанша, сол процестің басқа жібі орындала алады. Жіптер рпроцессорды жоспарлаудың әр түрлі нұсқаларымен сәйкестікте процестер сияқты бөледі.

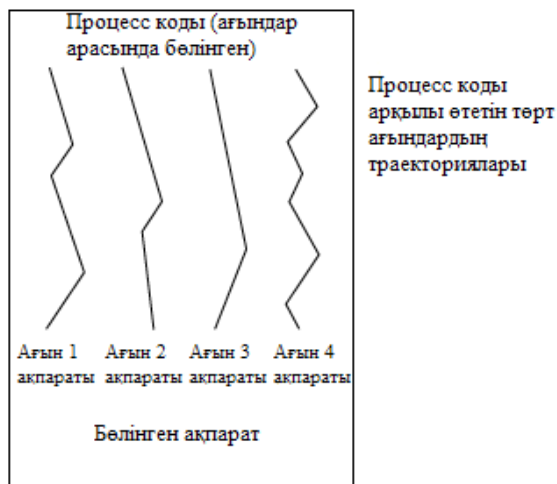
Бірақ бір процесс шеңберіндегі әр түрлі жіптер бқлек процестер сияқты соншама тәуелсіз емес. Барлық мұндай жіптерде бір және сол адрестік кеңістігі болады. Бұл олар бір жіне сол ғаламдық мәселелерге қосылатынын білдіреді. Әрбір жіп әрбір виртуалды адреске рұқсатқа ие болғандықтан, бір жіп басқа жіптің сүрлеуін қолдана алады. Жіптер арасында

толық қорғаныш жоқ, себебі, біріншіден, ол мүмкін емес, ал, екіншіден, керек те емес. Бір процестің барлық жіптері әрқашан бір пайдаланушының ортақ міндетін шешеді, сондықтан осында жіптердің аппараты міндеттің параллельдеу жолымен аса жылдам шешілуі үшін қолданылады. Адресітік кеңістікті бөлуден басқа, барлық жіптер, сонымен бірге, ашық файлдардың, таймерлердің, сигналдардың, т.б. жиыны бөледі.

Сонымен, жіптерге мыналар меншікті болады: бағдарламалық санауыш, сүрлеу, регистр, жіптер-буындар, қалып-күй.

Жіптер келесі түрлерге бөленеді: адресітік кеңістік, ғаламдық айнымалылар, ашық файлдар, таймерлер, семафорлар, статистикалық ақпаратты.

Көпжіпті өңдеу көпміндетті өңдеумен салыстырғанда жүйенің жұмыс істеу нәтижелілігін жоғарылатады. Мысалы, Windows көпмақсатты ортасында электрондық кестемен және мәтіндік редактормен бір уақытта жұмыс істеуге болады. Бірақ, егер пайдаланушы өзінің жұмысшы парағының қайта міндеттеуін сұраса, электрондық кесте көп уақытты қажет ететін осы амал аяқталғанға дейін құрсауланады. Көпжіпті ортада егер электрондық кесте бағдарламашыға ұсынылатын көпжіпті өңдеу мүмкіншіліктерін есепке ала отырып құрастырылған жағдайда, бұл мәселе туындамайды, сондықтан пайдаланушының әрқашан электрондық кестеге рұқсаты болады.



4.6-сурет Процесс жіптері.

жолы арқылы коды, дегенмен осы төрт жіптер ортақ кейбір деректер, олардың әрқайсысы өз бар жеке деректерді, сондай-ақ.

Жіптер қызмет етеді басқа да мақсаттарға саласындағы бағдарламалау. Көмегімен жіптерден процесінде, жіптер бақылай алады, олардың қол-ма-қол ақша және басқа жіптермен тігіледі. Қарастырайық процесс, ол құрамында екі ағыны. Бір ағыны жасайды маңызы бар қаланың үшін буфер жад. Басқа тұтынады маңызы бар буферінен. "көп ағымды, процессор өшеді арасындағы ағындарын. Дегенмен, лек-тұтынушы жалғасуы мүмкін емес, егер буфер бос болған. Бұл ештеңе жоқ, тұтынуға. Осылайша, тұтынушы жіп мүмкін өзін итеріп күте өндіруші stream дегенді білдірмейді жаңа нүкте байланыстыру қол жетімді болды. Екінші жағынан, өндіруші жіп мүмкін мәжбүрлеу керек, егер буфер емес толтырылған, содан кейін, тұтыну сигналдар, ол потреблял нәрсе. Екі ағынының мүмкін "кездесу" кездесуге. Мәжбүрлеу керек, ағындары бұл мысалда пайдаланады кооперативтік көп мағыналық (көп ағымдық), олар өз еркімен бас тарту процессор емес, мүмкін бас тартуға мәжбүр процессор кезде, таймер жұмысқа алмаса.

Көшу бір қадам алға қарағанда, кездесуге, онда маңызды екі жіптер немесе процестер тырысуда емес, қол жеткізу жалпы тіректік нүктесінен сол уақытта. Қарастырайық, бұл орын алуы мүмкін, егер ағыны 1 тырысады қол жеткізу жалпы буфер ағыны үзіледі ауысу контекст. Ол көшірді келесі элементі буфер, бірақ мүмкіндік болмады алып тастау buf - карта парағы көрсету үшін, нақты элемент қазіргі уақытта "калория". Енді, поток 2, жаңартылады, ол тырысады орналастыру тірек нүктесін буферде ғана табу үшін, буфер толы. Бұл, шын мәнінде, толық емес, өйткені бір тармағы тұтынылды. Осылайша, қол жеткізу буферге ағынымен 2 туғызады ағынын күтіп, ол емес тиіс. Басқа жағдай, екі лектің бірдей нүктесіне байланыстыру. Естеріңізге сала кетейік, біздің банктік мысал бұрын осы тарауда, келіңіздер пайдалану өзара айрықша қол жеткізу. Ағындар үшін, тағы да, қол жеткізу жалпы деректі тиіс өзара ерекше. Енді біз қарастырамыз сол мысал пайдалана отырып, ағындар. 1 тақырып оқиды тексеру шот балансының мөлшерінде \$ 1000. Ағым 1 дайын шегеруге \$ 1000, оған, бірақ үзіледі таймером. 2-тақырып оқиды баланс тексеру шоттар, \$ 1000, \$ 1000 шегереді, одан сақтайды және жаңа мән кері, \$ 0. Ауысу кері жіп 1, ол ескірген мәні, \$ 1000.

Ол шегереді \$ 1000, \$ 0 дүкендер бұрын. Нәтижесі болып табылады мәні орнатылған тең \$ 0, бірақ екі түрлі жіптер вычтены \$ 1000. Бұл мүмкіндік зақым келтіруі. Қамтамасыз ету үшін тиісті кіру жалпы деректерге, операциялық жүйе қолдайтын дұрыс қадамдастыруға болады.

Синхрондау талап етеді процесс немесе ағыны үзілмейді кезінде қол

жеткізуге тірек нүктенің немесе нүкте байланыстыру қол жетімді емес, тек бір процесс немесе ағыны бастады, оған қол жеткізу, ол босатылды. Мультибағдарлау, көп мәселелік және көп потокты талап етеді синхрондау.

Артықшылығы көп ағынды OS үстінен көпмағыналы ОС болып табылады контекст ағындары арасындағы жүреді, тез және тиімді қарағанда, арасында ауысу процестері. Бұл себебі болып табылады, бұл ауысу контекст талап етеді аз санын операцияларды сақтау үшін ағымдағы ағыны және қалпына келтіру ағынның қарағанда, бұл үшін сақтау және қалпына келтіру процестері.Елестетіп көріңізші сүйеніп, кері-суретте 4.5 коды осы процестің P7 болды ауытыс жадынан және кері виртуалды жад (своп). Ауысу контекст арасындағы P3 P7 және бірнеше апаттық, өйткені операциялық жүйе болады ғана емес, мәртебесі дүкен P3 және қалпына келтіру мәртебесі P7 емес, тура жүктеу бірнеше беттен P7 дисктен өтті. Болмайды ауыстырғанда арасындағы ағындарын. Болжаймыз, не P3 және P7 жіптер бір және сол процесс. Ауысу контекст олардың арасындағы әлдеқайда аз күш, сондықтан әлдеқайда жылдам.

Көптеген компьютерлер, тіпті бертінге дейін болды ретінде белгілі бір процессорлық жүйелер. Яғни, олар тек бір процессор. Бір процессор тек бір процесс немесе ағыны болуы мүмкін белсенді түрде орындау кез келген уақытта. "мультибағдарлау, көп мағыналы және көп ағынды өңдеу, бірнеше процестерін (немесе ағындар) болуы мүмкін белсенді, бірақ процессор болады орындау тек бір уақытта кез-келген. Осылайша, параллель орындалуы мүмкін болды арқылы кез келген осы жолдардың процесін орындау, бірақ шындыққа сәйкес параллельді өңдеу. "Көп процесті, компьютер бар бір процессор. Егер компьютер, мысалы, екі процессор, ол орындай алады екі процесс немесе ағынының бір мезгілде, бір-бірден әрбір процессор. Бүгін көптеген компьютерлер жүргізіледі бірнеше процессорлары.

Неғұрлым дәстүрлі нысаны көппроцесті жүйесінің бірі, ол тармақшасында, сатып алынатын тауарларға бір чип процессор. Мұндай компьютерлер қымбат және бірнеше дербес компьютерлер болды multiprocessor - жүйесін. Дегенмен, одан әрі милитаризация көрген соңғы онжылдықта, жеткілікті орын берілді бір процессордың орналастыру үшін бірнеше ядролардың. Осылайша, қазіргі заманғы процессорлар деп аталады қосарланған немесе төрт ядролы процессорлары. Әрбір өзек, шын мәнінде, процессор; ол өзінің ALU (арифметика / логикалық құрылғы), басқару блогы, регистрлер және кэш. Жалғыз, ол бөліседі басқа процессорлары чипіне болып табылады түйреуіштер, олар бекітіледі, бүкіл жинағы өзектердің жүйелі құрсымына компьютердің. Параллель өңдеу

мүмкіндігі болып табылады және қоңырау шалу. Мүмкіндігі жасалады орындауда бірнеше процестерді бір мезгілде. Мысалы, бір ядро орындай алады процесі веб-шолғыш болса, басқа орындай алады бағдарламасы, ойын-сауық, осындай бейне немесе mp3-плеер. Мәселе, алайда, қамтамасыз етуден тұрады қатар әрбір ядро пайдаланылады барынша мүмкін дәрежеде. Егер пайдаланушы тек бір жұмыс істейді процесс, бұл процесс қандай да бір жолмен болуы мүмкін бөлінген бірнеше өзектердің осылайша, әрбір ядро орындайды жеке бөлігі. Бұл талап өзге де ойлау жазу кезінде бағдарламалық код. Бағдарламашы мүмкін көріңіз бөліп логикасын бағдарлама компьютерлік графика кіші. Осылайша, бір ядро орындай алады логикасын және басқа ядро компьютерлік графика кіші. Идеясы тарату өңдеу үшін бірнеше процессорлар деп аталады бағдарламалаумен қатар, ол қалып отыр бір үлкен мәселелердің ашық бағдарламашылар үшін.

Бұл бөлімде қаралады және салыстырылады кейбір әдістері басқару процесін. Елестетіп көріңізші, сіз іске қосу үш бағдарлама деп аталатын, p1, p2, p3. Біз көреміз, олар қалай жұмыс бір жүйеге қойылған міндеттерді пакеттік жүйесін, мультибағдарлау жүйесін және көп мәселелік.

Бір міндеттерді қою, p1 орындалады болғанша емес, аяқталады, содан кейін p2 іске қосылады, ал содан кейін p3 іске қосылады. Бұл өте жағымды пайдаланушы үшін емес, өйткені жасауға қабілетті бірнеше заттарды бір мезгілде (мысалы, сіз редакциялау құжат, Интернеттен іздеу үшін кейбір ЗАЙДЕЛЬ, жауап беру және электрондық пошта арқылы, сіз оларды бір мезгілде, сіз олардың барлық жұмыс істейді, сол уақытта). Сіз көшіру және қою бір бағдарламаның басқа не, себебі бір бағдарлама басталғанға дейін аяқталады.

Бұл пакеттік өңдеу, орындау үш процестерді іс жүзінде болғандықтан бір міндеттерді қою. Бар өзгерістер. Біріншіден, өйткені пакеттік өңдеу кез келген бас тартады өзара іс-қимыл пайдаланушы, барлық кіріс мәліметтер жеткізілуі тиіс бағдарламасымен және кез келген шығу жүзеге асырылады орындағаннан кейін ғана аяқталады. Екіншіден, жеке өңдеу жүйесі үшін пайдаланылуы мүмкін жоспарлау үш процесс, олар қосылады кезекте күту. Жоспарлау болуы мүмкін бірінші келді-бірінші қызмет көрсетілді, басымдық, ең қысқа жұмыс бірінші немесе қандай да бір басқа нысан.

"Көп мәселелік, компьютер жұмыс істейді, осы уақытқа дейін не p1 p1 емес, тоқтатылды немесе p1 орындау үшін қажетті енгізу / шығару. Осы сәтте, операциялық жүйе қозғалу p1 кезекте күту және бастау p2. Енді p2 болады, осы уақытқа дейін не қажеттілігін p2 I / O, p2 тоқтатылмайды, немесе p1 жаңартуға дайын. Егер p2 талап етеді енгізу / шығару

немесе тоқтатылады, онда операциялық жүйе бастайды р3. Егер р1 дайын болады, р1 жаңартылады және р2 күтуге мәжбүр. Ал мультибағдарлау, енгізу және шығару жасалуы мүмкін талабы бойынша қарағанда, партия, көп мәселелік бере алмайды пайдаланушыға сыртқы түрі өзара іс-қимыл компьютермен, процестер, мәжбүр да орталық процессор және енгізу-шығару құрылғыларымен.

"Көп мәселелік, біз шын мәнінде аламыз жабуды орындау, сондықтан қолданушы айта алмайды процессор ауысады процесстер арасында. Үшін көп мәселелік процесі келесі түрде жұмыс істейді:

- жүктеме р1, р2, р3 естелік (немесе басым бөлігі әрбір процесінің қажеттілігіне қарай)
- Қайталау
- Start (немесе түйіндеме) р1 және орнату таймер жүйелі берілген көлемін (мысалы, 10000 циклдар)
- Декремент таймер кейін әрбір машиналық цикл
- Егер таймер жетеді 0 шақырыңыз, ОС
- Операциялық жүйе орындайды ауысу контекст арасындағы р1 және р2 - Ауысу контекст талап етеді, сақталу мәртебесі П1 және маңызы бар тіркелімдер және жай-күйі оралуына Р2 мәні мен тіркелімдер, мүмкін, сондай-ақ жаңарту жад қажеттілігіне қарай
- Бірге қайталаңыз р2
- Бірге қайталаңыз р3
- Еткенге дейін барлық процесстер тоқтатылмайды

Егер қандай да бір р1, р2, р3, және жіптің бір процесс болса, онда көпағымды болады ұқсайды қоспағанда ауысу контекст аз уақыт болады.

Буттинг және старт жүйесі

Негізгі жады летуч, яғни, ол қажет тұрақты қорек беруді, оны сақтау үшін мазмұн. Қоректену жүйесін өшіріңіздер және негізгі жады айналады бос (оның барлық мазмұн айналады 0s). Жұмыс аяқталғаннан кейін компьютерді төмен, жады ештеңе жоқ. Келесі жолы, сіз қосасыз компьютер жады бос болып қалады. Бұл өте маңызды, себебі, пайдалану үшін компьютер қажет: ОС жүктеледі жады жұмыс істейді. Операциялық жүйе-бұл бағдарлама, ол мүмкіндік береді жүктеп алуға және басқа да бағдарламаларды іске. Бұл парадоксалды жағдайды жасайды қалай аламыз жүктеу операциялық жүйесі жұмыс істейді, ол операци-

ялық жүйе, ол қамқорлық жүктеу бағдарламаларының орындалуы үшін ОЖ емес, жад, біз алдымен компьютерді іске қосыңыз? Бізге арнайы, бір реттік процесс, деп аталатын жүктеуді. *

Жүктеу процесі келесі түрде жұмыс істейді:

1. Процессорды жүктейді жібереді басқарушы сигналдар әр түрлі компоненттері компьютерлік жүйесі.
2. Енгізу / шығарудың базалық жүйесі (BIOS) орындайды қосу тамақтану өзін-өзі (POST), онда ол тексереді, жад құрылымы және тесттер, жауаптар. Сондай-ақ, келесі міндеттерді орындауға қалай орнату жүйелік сағаттар, іске қосу немесе өшіру, түрлі компоненттері, сондай-ақ өзара бақылаушылармен дискілер.
3. Контроллерлер қатты дисктердің (SCSI, содан кейін IDE) түсіндіруге дабыл береді өзіне және басқа да құрылғылар тестіленеді, желілік карта, USB және т. б.
4. BIOS анықтайды, қайда сақталады? Әдетте, бұл тестілеу арқылы әрбір осы құрылғылардың белгіленген тәртіппен іздестіру ОС дейін, табылды: дискеттер, CD-ROM, бірінші диск, екінші дискіге немесе желі.
5. Егер сіздің компьютерге бірнеше операциялық жүйелерді, онда бағдарлама тиеушінің орындалады анықтау үшін қандай операциялық жүйені жүктеу үшін және іске қосу. Екі бағдарлама үшін пайдаланылатын қос жүктеу Windows, Linux болып табылады GRUB (Grand Unified Тиеуші) және LILO (Linux Loader).
6. Ядро ОС тиеледі дисктен өтті. Ядро білдіреді бөлігінде операциялық жүйе, ол болуы тиіс есте сақтау, пайдалану үшін компьютер немесе негізгі компоненттері мен операциялық жүйе.
7. Қазіргі уақытта, басқару ауысуда бағдарламасын жүктеу ядросы операциялық жүйе, содан кейін іске қосады скриптер қосуға үдерісін аяқтау үшін жүктеу. Скриптер қосуға дәл олар естіледі ретінде, сценарийлер қабығы, ол іске қосылғанда, жүктейді түрлі аспектілері ОС.

* Термин тиеу және қалай сөздер өзін орнату, ол сипаттайды, қалай тартып шығарыңыз да, өзіне өзінің меншікті белдіктерімен boot-, өрт, ол туындайды түн ортасында да бар өрт дабыл қағуда. Өзі термин барон Мюнхгаузенаның айтуынша.

(а) Linux басталып, іске қосу жолымен бағдарламаны баптандыру. Оның негізгі жұмыс, оның ішінде іске қосу үшін ОС белгілі бір деңгейде іске қосу, мысалы, тек мәтін, мәтін желісімен, графикамен немесе графикалық жүктемесі-желі. Кейін таңдалған деңгей бағдарламасы іске қосады басқа сценарийлері сияқты сценарийлер `rc.sysinit`. Бұл сценарий негіз-

делген деңгейде іске, іске қосады, әр түрлі қызметтер. Мысалы, егер жүйе іске қосылады деңгейінде желісіне қол жеткізуді, онда желілік қызметінің тиіс іске қосылды.

(б) Windows операциялық жүйесінде, шешімдердің бірі болып табылады іске ОС қауіпсіз режимінде немесе толық экран режимінде пайдаланушының. Қауіпсіз режимін білдіреді диагностикалық режимі негізінен үшін пайдаланылады сонымен қатар, жүйелік әкімші жоя зиянды бағдарламалық қамтамасыз ету бағдарламалық қамтамасыз ету жоқ, бұл мыс жылдардың өзі немесе жылжыту. Тағы бір міндет баптандыру анықтау үшін, егер қауіпсіз Кіру қажет және ол ашу үшін журнал терезесінде. Бірі соңғы қадам баптандыру іске қосу үшін қабығы нақты пайдаланушының негізге ала отырып, оның кім кірдіңіз.

Жүктеу процесі бағдарламасын білдіреді. Дегенмен, ол үнемі жедел жады, себебі, оны компьютерге қосылған, ОЗУ бос. Өйткені жүктеу процесі өзгермейді, біз сақтауға, оны ТЕСҚ болып табылатын энерготәуелсіздік, өйткені оның мазмұны берік тіркелген. Дегенмен де, ПЗУ, әдетте, қымбат, өйткені бөлігін тиеу процесін иілгіш болуы тиіс, біз сақтауға порция бағдарламасын жүктеу бағдарламасы сияқты тиеушінің бөлігінде және BIOS, қатты дискідегі.

Тіркелім әкімшілігі

Орындау үшін жүйелі операцияларды бағдарланған көптеген операциялық жүйелер бар жинағы артықшылықты нұсқаулықтармен. Мысалы, жасау үшін жаңа пайдаланушы тіркелгілерін, тексеру болуы дыр қауіпсіздігі және айла-шарғы емес, жүйелік файлдармен болуы қажет admin кіру.Көптеген операциялық жүйелердің екі түрі бар есептік жазбаларды, тіркелгілері мен тіркелгі әкімшілерінің. * Жазбасы әкімшісінің кейде деп атайды мың немесе суперпайдаланушы. Тіркелгі әкімшісінің жеткізіледі, құпия сөз әкімшісінің, санаулы ғана білуі тиіс болдырмау үшін кездейсоқ пайдаланушыға жүйеге кіру әкімшісі ретінде және бір нәрсе, бұл жасалуы тиіс емес.

Windows операциялық жүйесінде есептік жазба әкімшісі деп аталады. Ауысу үшін әкімшінің, сіз жүйеге кіру әкімшісі ретінде енгізіп, пароль дұрыс. Linux, есептік жазба деп аталады, мың. Өзгерту үшін - өсіп тұрған, сіз командасын SU (ауыстыру пайдаланушы). Әдетте, су үшін пайдаланылады Көптеген операциялық жүйелердің екі түрі бар есептік жазбаларды, тіркелгілері мен тіркелгі әкімшілерінің. * Жазбасы

әкімшісінің кейде деп атайды мың немесе суперпайдаланушы. Тіркелгі әкімшісінің жеткізіледі, құпия сөз әкімшісінің, санаулы ғана білуі тиіс болдырмау үшін кездейсоқ пайдаланушыға жүйеге кіру әкімшісі ретінде және бір нәрсе, бұл жасалуы тиіс емес.

Windows операциялық жүйесінде есептік жазба әкімшісі деп аталады. Ауысу үшін әкімшінің, сіз жүйеге кіру әкімшісі ретінде енгізіп, пароль дұрыс. Linux, есептік жазба деп аталады, мың. Өзгерту үшін - өсіп тұрған, сіз командасын SU (ауыстыру пайдаланушы). Әдетте, су үшін пайдаланылады* Кейбір операциялық жүйелер бар, екі деңгейден астамды шоттар. Операциялық жүйе MULTICS, мысалы, бар сегіз деңгейлерін қол жеткізу, әрбір деңгейі неғұрлым жоғары барлық құқықтарды иеленеді қол жеткізу.

Қолданушы жазуға немесе оларды орындауға мүмкіндігі бар, бірақ басқалар да тек көруіне болады. Мысалы, сіз useradd және userdel іске қосу үшін суперпайдаланушы болуыңыз шарт, яғни, тек жүйелік әкімшілік (тер) пайдаланушылардың есептік жазбаларын қосуға немесе жоюға мүмкіндігі бар. Linux-тегі түпкі есептік жазбасына да сондай-ақ, ол да осы түпкіліктің үй бағыттылығында осы бөлімде болмауы керек. Ереже бойынша пайдаланушылардың үй каталогтері, /үйде/ пайдаланушы аты, сондай-ақ түпкілікте/ түпкілік ретінде болады.

АТ-адам ретінде, сіз үнемі әкімшісі ретінде жүйеге кіргеніңізді және одан шыққаныңызға хабардар болуыңыз керек. Linux-те сіз екі тәсілді ажырата аласыз: командалық жолда жолды қарауды. Linux-те, түпкілік жолы, ереже бойынша, әдетте # және тез қолданушы, әдетте, \$ (егер сіз оны өзгертпесеңіз). Сіз сондай-ақ Linux-те кім екендігіңізді WHOAMI командасы көмегімен біле аласыз. Сіз кейбір команаларды, түпкілік сияқты кенеттен бергіңіз келмейтіндіктен, кім екендігіңізді есте сақтау маңызды. Мысалы, сіз пайдаланушысыз, және сіз каталогтағы кішікаталогтарды қосқанда, барлық файлдарды, жойғыңыз келеді делік, онда сіз осы каталогке ауысып және -rf *-ты шағарасыз. Бұл дегеніміз - «рұқсат сұрамастан рекурсиялы түрде барлық файлдарды жою» дегенді білдіреді. "Рекурсиялы" деген сөздің астарында файлдарды Жою, сондай-ақ, ішкікаталогтардың барлығын және оның файлдарын жою деген сөз жатыр. Егер сіз дұрыс емес каталогта болсаңыз, операциялық жүйе, бәлкім, сізге файлдың иесі оның орындау жағдайының болмай тұрғанын айтуы мүмкін. Егер сіз дұрыс емес каталогта болсаңыз және де сіз түпкілік болсаңыз, онда операциялық жүйе жоюды орындайды және сіз қателесіп дұрыс емес файлдарды өшіресіз. Сіз еске түсірместен ешқандай мәселем жоқ деп ойлауыңыз мүмкін, дегенмен сіз ол жерде түпкі ретінде бір терезеге кіресіз, ал басқа сәтте өзінді өзгерту үшін

операциялық жүйенің (түпкі ретінде) өзгерту теңшеліміне кіресіз және (әдеттегідей қолданушыдан тыс) ретінде бұл өзгерістерді тексересіз.

Операциялық жүйені орнату

Бүгінгі таңда көптеген компьютерлерді сатып алу кезінде, оларда алдын-ала операциялық жүйе орнатылған болады. Кейбір пайдаланушылар, компьютерлік вирус, өшірілген файлдар немесе ескірген операциялық жүйе сияқты әртүрлі жағдайларға байланысты бәлкім, басқа операциялық жүйені қаласа, онда операциялық жүйені қайта орната немесе жаңартып алды. Сонымен қатар, операциялық жүйені қосу- яғни, ағымдағы операциялық жүйе ауыстырылған немесе жойылған екендігін білдірмейді.

Қолданушы бірнеше операциялық жүйелерді орната алады, әрқайсысы өз диск бөлімде, сондай-ақ GRUB, LILO және BOOTMGR сияқты жүктеу бағдарламасын қолдана алады. Компьютер алғашқы қосылған кезде аппараттық қамтамасыз етудің төрт баптандыру кезеңі орындалады (жүйені жүктеуді және баптандыру бөлімін қараңыз). 5-қадам - ОЖ жүктеу үшін пайдаланушыға нұсқау ретінде қамтамасыз ететін жүктеушінің бағдарламалары орындауы болып табылады. Содан кейін таңдалған операциялық жүйеде кейін бапталатын таңдау орындалғаннан кейін, пайдаланушы таңдалған ОЖ-де компьютерді қолдана алады. Операциялық жүйені өзгерту үшін, пайдаланушы бастапқы жүктеуші бағдарламаларына қол жеткізу үшін ағымдағы операциялық жүйені жабуы және қайта іске қосуы керек. Компьютерде бірнеше операциялық жүйелерді жүктеу қиындықтарға әкелуі мүмкін, әсіресе операциялық жүйелердегі жаңарту немесе қатты дискті қайта бөлу кездерінде. Басқа тәсіл ретінде бірнеше ОЖ-ге кіру мүмкіндігі виртуалдық машиналарды қолдану болып табылады. VM, өзінің мәні бойынша өзіне жеткілікті орта болып табылады, онда сіз ОЖ-ні орната аласыз. Виртуалды машинаның өзі қатты дискіде мәліметтер түрінде орындалғанша сақталады. Осылайша, виртуалды машина дискілік кеңістікті алады. Егер сіздің компьютерде процессор болса, онда VM орындау компьютеріңізде айтарлықтай бәсеңдетуге әкеліп соқтырады, бірақ заманауи көп ядролы процессорды қолдану арқылы виртуалды машиналар тиімді жұмыс істеуге мүмкіндік береді. Бұл сондай-ақ екі (немесе одан көп) ОЖ ашылуына және виртуалды машинада терезелерінен басқаларға жай ғана ауысуға мүмкіндік береді. VM бағдарламалық қамтамасыз ету көмегімен, сіз бірнеше виртуалды машиналарды және кез келгенін немесе бір уақытта барлығын жасай аласыз. Бұл бөлімнің қалған бөлі-

гінде екі операциялық жүйе Red Hat Linux (Ерекше түрде, CentOS 5.5), және Windows 7 қалай құру керек екендігі сипатталады. Бұл әрекеттенуде немесе монтаждау кезінде, VM ішінен сіздің не істейтініңіз ұсынылады. Коммерциялық және тегін VM бағдарламалық өнімдер бар, осы сияқты VMWare клиент VMware, VMware Player, VMware Server және Vsphere және VirtualBox Sun (немесе VBox).

Windows-ды орнату

Windows 7-і орнату үшін сіз алдымен, Windows 7 дискісін компакт-дискке салып содан кейін барып жүктеуді бастайсыз. Сіздің компьютерінің қатты дискке ОЖ-ні орнатуға дайын бірақ, сізде алдын ала дайындалған ОЖ болмаса онда қалыпты жүктеу процесін тоқтатасыз. Бұл жүктеуде F12. * функциялық пернені басу арқылы орындалады. Сіздің компьютеріңіз жауап бергенше пернені қайта-қайта басыңыз. Бұл жүктеу яғни, әртүрлі құралғы тізімі бар, өзіңіздің қалауыңыз бойынша орнатылатын, ереже бойынша қатты диск, оптикалық диск, желі, USB-құрылғы, және дискет параметріне әкеліп соқтырады. Оптикалық дискті таңдаңыз. Оптикалық диск колжетімді және орнату процесі басталады.

Бірнеше минуттан кейін, сізге бірінші бірнеше Windows 7 диалогтық қондырғылары көрсетіледі. Типтік орнату жүктеу кезінде әрбір терезе сайын «Әрі қарай» пернесін басуды талап етеді. Ең басында сізге тіл орнату үшін (мысалы, ағылшын тілінде), уақыты және валюта формат, сондай-ақ іске қосу / пернетақта типі (мысалы, АҚШ) таңдау ұсынылады. Содан кейін Install Now пернесін басу сұралады (ол жалғастыру үшін жалғыз нұсқа). Сізден лицензиялау шарттарын қабылдау сұралады.

Содан кейін Сізге іріктеп орнату немесе жаңарту үшін опциялар беріледі. Сіз жаңартуды таңдасаңыз, егер де сізде қазірдің өзінде Windows 7 нұсқасы орнатылғанын көрсеңіз, жүйені жетілдіресіз. Бұл, мысалы, сізде Windows 7 ішінара бүлінген немесе жылы ескірген кезде жасалады. Таңдамалы орнату алдыңғы барлық файлдарды, параметрлерді және бағдарламаларды сақтамайды, яғни жаңартуда барлығы сақталады. Пайдаланушылық орнату сондай-ақ, дискілерді өзгертуге, бөлуге мүмкіндік береді. Содан кейін Сіз терезе қайда орнатылуы тиіс екендігін сұрайсыз. Егер де сізде тек бір қатты диск қана болса, онда жасау үшін ешқандай таңдау жоқ.

Осы кезден бастап бағдарламаны орнату біраз уақыт үзбей немесе қажет болған пайдаланушылық іске қосу (мүмкіндігінше, 10 минуттан 20 минутке дейін, сіздің оптикалық жетек және процессорыңыздың жыл-

дамдығына байланысты) жұмыс істейтін болады. Бұл уақытта, Windows бірнеше рет қайта жүктелетін болады. Қайтадан сұрау кезінде компьютердің атын және бастапқы қолданушының есептік жазбасын құру сұралады. Компьютердің әдепкі аты - есептік жазбаның аты, сіз оны, -PC арқылы кейіннен енгізесіз. Мысалы, егер сіз Zappa атын көрсетсеңіз, онда компьютер әдепкі Zappa-PC болады. Сіз, әрине, әдепкі атауын өзгерте аласыз. Келесі терезеде, есептік жазбаның мәліметтерін аяқтап, есептік жазба үшін бастапқы құпиясөз «Ишара» болады, бұл егер де сіз құпиясөзді есіңізден үнемі шағарып алатын жағдайда болсаңыз. Бұл бастапқы шот әкімшіден есептік жазбаны құруын талап етпей, пайдаланушыға компьютерді бірден бастау үшін мүмкіндік береді. Нақты функционалдық перне нақты өндірушіге байланысты ерекшеленуі мүмкін. Нұсқаулық пайдалану үшін қандай функционалдық перненің керек екендігін хабарлау үшін жүйенің жүктеуі уақытында көрінеді. Жалғастырмас бұрын, Windows қазіргі уақытта өнім кілтін сұратады. Бұл код, CD шебер орамда көрсетілген. Кілт әріптер мен сандардан және ұзындығы 25 символдан тұратын комбинация болып табылады. Бұл сіздің Windows, оның құқығы бар және бұзылмауына кепілдік береді.

Келесі қадам Windows үшін, автоматты жаңартуларды орындау үшін. Дегенмен бұл қадам міндетті емес, бұл өте пайдалы болып табылады. Бұл сізге қондырғылар алу үшін ең соңғы патч Windows 7 операциялық жүйесіне мүмкіндік береді. Бұл қадамсыз, сіз шектелген ОС нұскасын орнатуға шектелесіз, яғни ол кезде компакт-диск дайындалған болатын. Егер сіз осы қадамды өткізсеңіз де, Windows-да жаңарту орнату кейінірек уақытта болады, мысалы, бірінші рет компьютерді қосуға тырысасыз. Сіздің нұсқаларыңыз: үй желісі, желілер мен жалпы пайдаланулар. Бұл нұсқа, кейінірек уақытта алып тастау болып табылады. Таңдағаннан кейін, сіздің компьютеріңіз желіге қосылуға тырысады. Және, ақыр соңында, сіздің жұмыс үстеліңіз дайындалып, пайдаланушы режимде операциялық жүйе жүйелендіріледі. Сіз жұмыс істеуге дайынсыз!

Дегенмен сіз компьютерді пайдалануға дайынсыз, Windows бағдарламашылары орнатқан Windows параметрлерімен жүктеледі. Қазіргі уақытта, сіз жұмыс үстеліңізге өзгерістерді басқару панелі арқылы енгізе аласыз. Сіз, мысалы, терезелер стилін, жұмыс үстелінің фонын, түсін, пайдаланылатын параметрлерін, экранның рұқсатын, сондай-ақ жұмыс үстеліндегі белгішелер мөлшерінін өзгерте аласыз. Сондай-ақ жұмыс үстелінің төменгі бойындағы есептер тақтасына қандай бағдарламалар бекітілуі керектігін, сондай-ақ, жоғарғы жағындағы бағдарламалар мәзірінде не болу керектігін көрсете, қоя аласыз. Сіз сіздің желілік бран-

дмауэріңіз жұмыс істейтіндігіне көз жеткізуіңіз керек.

Windows 7-ні орнатуға өте көп уақыт кетпейді және оны орнату оңай. Басынан аяғына дейін барлық қондырғылар сағаттан кем емес, бәлкім, барлығы 30 минутты алады.

Linux Орнату

Windows 7 орнату, Red Hat Linux орнату компакт-диск көмегімен жасалуы мүмкін. Мұнда біз сізде нұсқаушы компакт-диск CentOS 5.5 бар екендігін болжаймыз. Windows-ға қарағанда, орнату қиын, және диск бөліктеулері сияқты түсініктерді талап етеді (5-тарауда диск бөліктеулері сипатталған).

Компакт-дискті орнату кезінде сізге экран ұсынылады, онда орнатудың бірнеше нұсқалары бар, мысалы, бұқаралық ақпарат құралдарын (шын мәнінде, егер сіз компакт-дискті меншікті өзіңізге құрмасаңыз қажет емес) және тіл параметрлерін орнату. Дисктің бөліктеу қадамына сіз енді жеттіңіз. Сірә, жаңа қондыруда, қатты диск әбден бөлімдерге не сындырылған емес, және сіз бөлімді көрсетуіңіз керек. Егер сіз ОС машинасында Linux-ты орнатсаңыз, сіз сақ болуыңыз керек, өйткені бұл дисктің бөлімі азат болуы керек. Бұл міндетті түрде екі не одан да көп қатты дискті талап етпейтіндігіне көңіл аударыңыз. "Еркін диск" логикалық тағайындау болып табылады және басқа ОС бөліктері болып табылмайды. Егер де сіз жойғыңыз келетін бөліктеулер болса, онда «таңдалған дисктердегі Linux барлық бөлімдерін жою және макет құру» және «Шолу және макетті бөлуді өзгерту» опцияларын таңдайсыз.

Мұнда сіздің Linux дискіне қажет бөлімдерді нұсқау керек. Сіз түпкі бөлімдердің әрқайсысына әртүрлі бөлімдерді, үстемелеуді, сонымен қатар, қолданушылар каталогтерін қалайсыз ба? Сіз неғұрлым жұқа топ бөлімдері арқылы, мысалы, / var және / usr үшін бөлімдер бар бола тұра; керісінше жағдайда, бұл каталогтар түпкілікке жіберіледі. Әрбір бөлім үшін, сіз құрастыру (каталог) немесе файлдық жүйесін (үстемелеу үшін), бөлімнің көлемін, және де бөлім көлемде құрылуы мүмкін бе немесе қалған қолжетімді кеңістікті толтыру нүктесін таңдайсыз. Мысал ретінде Linux дискін келесідей бөлуге болады:

- Құрастыру нүктесі: таңдаңыз / (түпкі), мөлшері-4000 (яғни, 4 ГБ құрайды), тіркелген мөлшерде
- Құрастыру нүктесі: таңдаңыз / Var, мөлшері 1000, тіркелген мөлшерде
- Файлдық жүйе типі: ауыстыру (құрастыру нүктелерін таңдамау), мөлшері 1000, тіркелген мөлшерде

- Құрастыру нүктесі: таңдаңыз / үйде барынша мөлшерін толтыру

Келесі экранда орнату бойынша сізге параметрлер ұсынылады. Әдепкі бойынша GRUB және таңдалған болуы тиіс және негізгі қатты дискке орнатылған ол, бәлкім, / DEV / sda1. Желілік құрылғылар үшін қандай IP-мекен-жайы қалай құрылуы тиіс. Ең көп таралған техника DHCP арқылы сервермен белгіленеді. Соңғы сұрақ, уақыт белдеуін таңдау үшін?

Сізден жүйенің түпкі құпиясөзін белгілеу сұралады. Бұл құпиясөз жүйелік әкімшіліктің оқыс жұмыстары талап еткен жағдайларда, бағдарламалық қамтамасыз етуді құру және есептік жазбаны құру арқылы жүйелік әкімші үшін үнемі қолданылып отырады. Сіз өзіңіз ұмытпайтын құпиясөзді қолданғыңыз келе ме? Келесі экранда сіз қандай бағдарламалық қамсыздандыру автоматты түрде CentOS-пен орындалуы керектігін көрсете аласыз. Desktop-Гном әлбетте орнатылған болуы керек. Қазіргі уақытта сіз басқа бағдарламалық қамтамасыз етуді таңдап, немесе кейін орната аласыз. Орнату дайын! Бұл процесс әдетте 5 минуттан 10 минуттық уақытты алады. Бұның барлығы орындалған соң, сізден жүйені қайта іске қосу сұралады. қайта іске қосқаннан соң, сіз орнату процесін аяқтайсыз. Бұл өзіне бастапқы брандмауэр теңшелімін қосады. Тағы да, әдепкі бойынша (орындау) жақсы. Сіз уақыт пен күнді құра аласыз. Және соңғы, Windows сияқты, сізден қолданушының бастапқы есептік жазбасын құру сұралады. Бұл есептік жазба қолданушы ретінде сіз графикалық интерфейске кіру үшін керек. Түпкі ретінде GUI-ға бір кездері кіруіңіз ұсынылмайды. Windows-ға қарағанда жүйемен жұмыс істеу үшін, сізге қазіргі мезетте компьютерді қайта іске қосу керек емес; оның орнына сіз пайдаланушының есептік жазбасы арқылы жүйеге кіріп, жұмысыңызды істеуіңізге әбден болады.

5. Файлдар, Тізімдемелер және файлдық жүйе

Бұл тарауда логикалық және физикалық файлдық жүйелерді іске асыру қарастырылады. Файлдық жүйенің логикалық көрінісі - файлды бірліктер ұйымдастығы, тізімдемелер және бөлімдер болып саналады. Логикалық көріністен файл компоненттерінің физикалық диск бөлігіне сақтауын қайта құру - физикалық іске асыру болып табылады. DOS/Windows сияқты файлдық жүйелер, сонымен қоса Linux тізімдемесінің жоғарғы деңгейін және блок индекстеуін іске асыруды ұсынады. Бұл тарау файлдық жүйеде командалық жол нұсқаулығының көмегімен қалай жылжуын зерттеумен аяқталады.

Бұл тараудың оқыту мақсаты:

- Файлдық жүйе элементтерін логикалық, сонымен қоса, физикалық тұрғыдан алып сипаттау;
- Файлдардың физикалық сақталу тәсілдері және олардың қол жетімді болуын сипаттау;
- Тізімдеменің жоғарғы деңгейін Windows-да да, Linux-та та қолдану көрінісі;
- Файлдық жүйе бойынша Linux және DOS командасының мысалдарын пайдаланып навигацияны суреттеу;

Бұл тарауда біз файлдық жүйенің компоненттерін және Windows файлдық жүйесін, Unix/Linux файлдық жүйесінде егжей-тегжейлі қарастырамыз.

Файлдар және тізімдемелер

Файлдық жүйеде абстракцияның екі деңгейі болады. Файлдық жүйенің физикалық табиғаты болады, бұл – деректерді сақтау құрылғысы, файл индексі, атаулар үрдісінің орналасуы және қатты диск ішіндегі

олардың орнын анықтау болып табылады (басқа да таратушылар). Сонымен қатар, файлдық жүйенің логикалық табиғаты да болады, ол да физикалық жүйе сияқты тәуелсіз атау нысандарын бөліп тастайды. Біз пайдаланушылар сияқты, жүйе әкімшісі сияқты да логикалық файлдық жүйеге қызығушылық білдіреміз. Біз файлдық жүйенің ақаулығын жоюға міндетті болған жағдайда ғана, бізге файлдар мен тізімдемелер бөлшектерінің қалай орналасқаны қызық болады (мысалы, зақымданған файлдар, файлдық жүйені дефрагменттау қажеттілігі). Ары қарай, біз бұл тарауда файлдық жүйенің физикалық табиғатын қысқаша қарастырымыз, ал қазір логикалық бағытқа көңіл аударамыз. Файлдық жүйеде нысанның үш түрі бар:

- Бөлімдер
- Тізімделер
- Файлдар

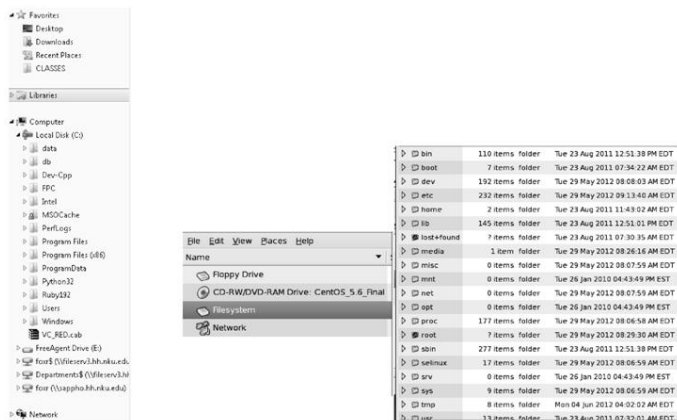
Файлдық жүйе шегінде бөлім логикалық бөлімше болып табылады. Windows операциялық жүйесінде бөлім әріп ретінде белгіленеді, мысалы, C:\ немесе D:\. Бұл екеуі де қатты дискқа жатады, бірақ екеуі екі түрлі бөлімге қатысты. Linux-та бөлім логикалық дисктің бөлек дискісіне жатады, бірақ шын мәнінде, бірнеше бөлімдер бір дискта бола бере алады. Кейбір бөлінген тізімделерді сақтай отырып, осылайша, біз бөлімде нақты тізімделер орнатамыз. Мәселен, жүктеу кеңістігі біріншіде, пайдаланушылардың үй тізімделері екіншіде, ал Linux (ОЖ) болса үшінде бөлімде орналасуы мүмкін. Әдетте, сіз операциялық жүйені бірінші орнатқан кезде бөлімді құрасыз. Орналастыру жүзеге асқан соң, бөлімнің мөлшері мен көлемін өзгерту ыңғайсыз болады, өйткені бұл осы бөлімнің кеңістігін кішірейтуі мүмкін, ал бұл кішірею соңында осы бөлімдегі кейбір файлдардың ауысуына немесе өшірілуіне әкеліп соғуы мүмкін. Бөлімдер аса қатты қызық болмауы мүмкін, өйткені, сіз өзіңіздің ОЖ орнатып болғаннан кейін, сізге қайтадан бөлек бөлімдер жайлы алаңдамайсыз.

Тізімдемелер папка атауымен белгілі, олар пайдаланушыларға файлдардың ауысын байланыстыруға мүмкіндік береді. Бөлімдерге қарағанда, сізге канша тізімдеме керек, сонша тізімдеме құра аламыз. Тізімдемелер тізімделер ішінде орналаса алады, соның арқасында, ол пайдаланушыларға иерархиялық файлдар кеңістігін құруға мүмкіндік береді. Бұндай тізімделер әдетте ішкі тізімдеме болып ескеріледі. Иерархиялық файлдар кеңістігі әдетте «ағаш» кейіпінде беріледі. Ағаштың жоғарысында немесе түбінде бөлімдер болады. Төменде тізімдеменің жоғарғы

деңгейі болады. Ең төменінде файлдар мен ішкі тізімдемелер болады. Ағашта жапырақ төменгі жақта орналасқандай етіліп тармақталады. Бағдарламаларды диспетчерлеу файлдары көбінесе ағашты солдан оңға қарай суреттейді (сол жағында ағаштың түбірі бар); оң жақ бағытқа қозғалу арқылы ішкі тізімдемелер кеңейтіледі, ал жапырақтар (бөлек файлдар болып табылады) кез-келген тармақтың оң жақ соңында болады. 5.1 суреті әдеттегі файлдық кеңістікті суреттейді, ал ол болса файл диспетчерінің бағдармалық жасақтамасында көрсетіледі: Windows сол жақта, ал Linux оң жақта болады. Компьютер екі операциялық жүйеде де ең жоғарғы деңгей болып табылады, ал ол болса өз алдына әртүрлі файлдық кеңістік пен дисктерден тұрады. Заманауи компьютерлерде әдетте қатты диск (C:) және оптикалық диск (D:) таратылады. Онда қосымша дисктер, сонымен қоса, сыртқы қатты дискі, иілгіш дискі, USB флэш-дискі, желі бойынша қол жетімді файлдық сервер немесе ішкі қатты дискі шегіндегі бөлімдер атап көрсетілуі мүмкін. Бұл соңғы жағдайда, бұлардың бәрі Linux-те 5.1 суретінде суреттелгендей бір ішкі файлдық жүйе ретінде берілуі мүмкін. Linux файлдық жүйесінің жоғарғы деңгейі қайта орнатылған тізімдемелер ретінен тұрады. Windows операциялық жүйесінде файлдық жүйе әдетте, бастапқы Program Files (Windows 7 екі директориясындағы 32-разрядты және 64-разрядты бағдарламаларды белгілеу), Users (пайдаланушылардың үй тізімдемесі), Windows (ОЖ) тізімдемелерімен шектеледі. Біз бұл тарауда кейіннен әртүрлі тізімдемелерді пайдалануды егжей-тегжейлі үйренеміз.

Файлдар, тізімдемелер және файлдық жүйе.

127



Сурет 5.1 Типтік файлдық кеңістік

151

Файлдар файлдық жүйеде сақталудың мағыналы бірліктері болып табылады. Олар деректерді және бағдарламаларды өзінің құрамына кіргізеді, бірақ біз бағдарламаларды файл деректерінің арнайы түрлері деп көреміз. Файл деректері барлық көлемді және түрлі болады. Файл деректері ол - мәтіндік файлдар, пішімделген мәтіндік файлдар, екілік файлдар немесе ерекше шығуы және басқарылу нышандарын пайдаланатын файлдар болып табылады. Linux-та мәтіндік файлдар өте кеңінен таралған. Windows операциялық жүйесінде файл деректері әдетте Word (DOCX) және Excel (XLSX) сияқты бағдарламалармен құрылады. Файл деректері сонымен қоса, мультимедиялық файлдар, музыкалық және дыбыстық файлдар (WAV, MP3), суреттеме файлдардар (GIF, JPG, BMP), бейне файлдар (MPG, AVI, MOV) да бола алады.

Файлдардың атаулары болады. Ескі ОЖ файлдардың атаулары тек сегіз нышанға дейін ғана шектеулі болған, кейіннен қосымша арқылы кеңейтілген, бұлар үш нысаннан тұрған. Атаудың бөлінуден кеңейтілуі мерзімді болып табылады. Ескі ОЖ файлдардың атауларында қолданылған нышандар тек алфавиттік, сандық, асты сызылған және дефисті нышандармен шектелген. Бүгінде, файлдардың атаулары бос орынды және кейбір басқа тыныс белгілер нышанын қамтиды, бірақ кез-келген нышан қол жетімді бола бермейді. Кейбір ұзақ атаулар жиі пайдаланылады, өйткені атау өте сипаттамалы тұрғыда болуы керек. Кейбір пайдаланушылар кеңейтілуді елемейді, бірақ кеңейтілу өте маңызды мағынаға ие болады, өйткені олар пайдаланушыларға файлдың қандай түрге жататынын көрсетеді (қандай бағдарламалық жасақтамадан құрылғанын көрсетеді). Файлдың қалай ашылуын анықтау үшін, ОЖ файлдарды кеңейту әдетте жиі қолданылады. Мысалы, егер сіз файлдық деректері бар белгіде тінтуірді екі рет шертсеңіз, егер файлдың кеңейтілуі болса, сізге ОЖ автоматты түрде бағдарламалық жасақтамаға сәйкес ашып береді, ал кеңейтілу сәйкесінші бағдарламалық жасақтамамен салыстырылады. Файлдың түрінен және атауынан бөлек, файлдардың басқа сілтеме қасиеттері бар. Олардың файлдық жүйеде логикалық орналасуы көрсеткішпен белгіленеді. Яғни, файлдың дескрипторы кеңістіктегі файлдық жүйеде оның орнын құрайды. Мәселен, `\\Users\foxr\Менің құжаттарым\cit130\Ескертпе\ch7.docx` файлдық жүйесі файлдың физикалық орнын емес, логикалық бағдарын сипаттайды. Қосымша үрдіс логикалық орыннан физикалыққа ауысуы тиіс.

128

Ақпараттық технологиялар

Файлдың көлемі, соңғы өзгертулердің уақыты мен күні, иесі де пайдалы қасиеттер болып табылады. Операциялық жүйелерге байлынысты

файлдың құрылу уақыты, күні және тобы сондай-ақ жазылуы мүмкін. Сіз бұл қасиеттерді Windows-та таба аласыз, ол үшін тінтуірдің оң жақ батырмасын кез-келген файл атауына басып Windows бастаушысынан Қасиеттер командасын таңдау керек. Linux-та сіз бұл қасиеттерді, LS-l командасын орындау арқылы көре аласыз.

Файлдарды бір толассыз блок дискісінің кеңістігінде сақтау міндетті емес. Оның орнына, файлдар тиянақты көлемге бөлінген, олар блок атымен белгілі. Файлдар блоктары бір жолақта немесе секторда, сондай-ақ сол беткейде болуы үшін дискте жайыла алмайды. Оның орнына, блоктар бірнеше диск беткейінде бөлінуі мүмкін. Бұл осы тарауда кейіннен толық қарастырылады.

Ескі ОЖ көптеген командалар файлдық жүйені басқаруға бағытталған. ДОЖ операциялық жүйесі, мысалы, Дискті Операциялық Жүйе деген мағынаны білдіреді. Дисктік операциялық командадан бөлек, пайдаланушыға керегі тағы да басқа кейбір командалар болды. Бұл операциялық жүйе тек қана файлдық жүйедегі амалдарды орындайды дегенді білдірмейді, сонымен қоса, пайдаланушы командалар басқа да амал түрлерін шешкен. Екі MS-DOS және Linux ортақ командалары төменде көрсетілген. Сонымен қатар, Windows тінтуіріне арналған амалдар қарастырылған. Біз Linux және DOS командаларын файлдық жүйеде Ауысу бөлігінде толығырақ үйренеміз.

Linux:

- ls – бұл тізімдемедегі файлдардың және ішкі тізімдеменің тізімі
- mv – файлды немесе тізімдемені ауыстыру немесе атауын ауыстыру
- cp – файлдарды немесе папканы жаңа жерге көшіріп алу
- rm – файлды ауыстыру (өшіру)
- cat, азды-көпті – файлдың ішіндегісін экран бетіне шығару
- mkdir – жаңа (ішкі) тізімдеме құру
- rmdir – тізімдемені өшіру; DOS командасы үшін тізімдеме бос болуы керек
- dir – осы тізімдемедегі файлдардың және ішкі тізімдеменің тізімі
- move – файлды жаңа жерге ауыстыру немесе атауын ауыстыру
- copy – файлды жаңа жер көшіріп алу
- del – файлды ауыстыру (өшіру)
- type – файлдың ішіндегісін экран бетіне шығару
- mkdir - жаңа (ішкі) тізімдеме құру
- rmdir – тізімдемені өшіру; бұл команда үшін тізімдеме бос болуы керек

Windows ОЖ (Windows бастаушының көмегімен, 5.2 суретін қар.)

- Тізімдеменің ішіндегісін қарау үшін, тізімдеме атауын сол жақ жаяужолда шертіңіз; оның ішіндегісі оң жақ жаяужолда шығады, егер тізімдеме ең жоғарғы деңгейде болмаса, ізделіп отырған тізімдемеге жету үшін, сіз файлдық жүйедегі барлық бастапқы тізімдемелерді бұру керексіз.
- Файлды ауыстыру үшін оң жақ жаяужолдағы файл белгісін ағымдағы сол жақ жаяужолдағы папка директориясына апарыңыз.
- Файлды көшіріп алу үшін тінтуірдің оң жағымен файл иконасына шертіңіз, көшіруді таңдаңыз, сол жақ жаяужолдағы осы тізімдемені басып, жаңа тізімдемеге ауысыңыз, содан кейін, оң жақ жаяужолда тінтуірдің оң жақ батырмасын басып Қою командасын таңдаңыз.
- Файлдың атауын ауыстыру үшін тінтуірдің оң жағымен файл иконасына басыңыз, атауын ауыстыру командасын таңдаңыз (немесе файл иконасының атауына асықпай екі рет басыңыз), содан кейін жаңа атауды жазыңыз.
- Файлды (папканы) өшіру үшін белгіні қоқыс себетіне апарыңыз (тінтуірдің оң жағымен файл иконасына шертіңіз, Өшіруді таңдаңыз, немесе тінтуірдің сол жағымен файл иконасына шертіп, Өшіру батырмасын басыңыз), файлды біржола өшіру үшін, сізге себетті жеке өшіру керексіз.
- Жаңа директория (папка) құру үшін New Folder бастырмасын басыңыз, немесе тінтуірдің оң жақ батырмасымен оң жақ жаяужолды шертіңіз және New таңдаңыз, кейін Folder басыңыз, содан соң, папка пайда болғаннан кейін, папканы атауын жазыңыз.



- Файлдың ішіндегісін көрсету үшін, сіз файл белгісін екі рет шертіңіз, сонда ол файлдың кеңеюіне қатысты бағдарламалық жасақтаманы іске қосуы керек.

Назар аударыңыз! Windows-ғы файлдардың кеңейтілуі файлдар менеджерінде көріне алмайды, сондықтан сізге файлдардың кеңейтілгенін көру үшін оны қайта дұрыстап күйге келтіру керексіз (егер оларды көргіңіз келсе).

Файлдық жүйедегі кейбір үлкен файлдардың кеңістіктегі қол жетімдігіне әсерін кішірейту үшін, пайдаланушы *файлды сығымдау* командасын орындай алады. Файлды сығымдаудың екі формасы болады. Файлдарды *жоғалтусыз* сығымдау түрінде файл көлеміне қарай кішірейеді, сонда ол өзінің бастапқы қалпына қайта ашылған кезде, ондағы ешбір деректер жоғалмайды. Файлдарды жоғалтусыз сығымдауды орындауда көптеген алгоритмдер болады, олар әсіресе мәтіндік файлдарда жақсы жұмыс істейді. Файлдарды жоғалтусыз сығымдауда, сіз әдетте файлды ашпас бұрын, сол файл қол жетімді болуы керек (ерекшелік бар, FLAC дыбыстық сығымдау алгоритмі). Файлдарды жоғалтусыз сығымдаудың артықшылығы қазіргі кезде қолданылмайтын файлдардың әжептәуір кішірек көлемінде сақталатыны болып табылады. Файлдарды сығымдау және қайта ашу көп уақыт алады, бұл ең негізгі жетіспеушілік болып саналады.

Файлдарды *жоғалтумен* сығымдау файл көлемін кішірейту үшін, шын мәнінде кейбір деректерді шығарып тастау болып табылады. Көптеген ағын типті аудио және бейне формалары – бұл деректерді жоғалту файлы болады. Бұларға .avi, .jpg, .mov (бейне) және .wav, .mp3 (аудио) кіреді. Көптеген сурет сақтау формалары да файлдарды жоғалтумен сығымдауды қолданады, бұл дегеніміз .jpg. Егер сіз растрлық файлды (.bmp) .jpg пішінде салыстырсаңыз, онда сіз .bmp-нің үлкен (мүмкін әлдеқайда үлкен) тазалыққа ие болотынын көресіз. .Jpg файлы жақын тұрған пиксельдердің «бұзылуы» арқасында кеңістікті үнемдейді. .Gif пішіні кейбір түстерді шығарып тастамайды, сондықтан суретте стандартты тақтайшадағы түстер қолданылады және онды сіз қандай бір түстер анықтығын жоғалтқанын көре алмайсыз.

Сіздің қолыңызда файлдар жинағы болса, онда келесі кезекте олардың бәрін бірге мұрағатқа жинақтау әрекеті болып табылады. Бұл кейіннен Интернет арқылы оларды жүктеп/кері жүктеу, яки, сақтық көшірмеде

оларды орналастыру, не болмаса топ ретінде файлдардың жеңіл ауысына мүмкіндік береді. Файлдарды мұрағаттау кеңітілген, бұл кезде бағдарламашы ашық бастапқы кодты бағдарламамен бөліскісі келеді. Бастапқы коды бәлкім, өзіне көптеген, мүмкін, ондаған немесе жүздеген файлдарды кіргізеді. Файлдардың топтамасын мұрағаттай отыра, барлық бастапқы жинақты бір әрекет арқылы жүктеуге болады, сол кезде ол пайдаланушыдан файлдарды жеке-жеке жүктеуді қажет етпейді. Мұрағаттаудың кеңінен таралған ең басты түрі ол zip(зип) пішіні арқылы, бұл пішін файлдарды сығымдауды, және де мұрағаттауды да жүзеге асырады. Файлды «зиптаудан» айыру үшін, ең алдымен файлды ашу керек (unzip). Мұрағаттау(zip) және мұрағаттан шығару(unzip), сонымен қоса, Windows-тағы WinZip WinRAR, Linux-тағы Gzip және gunzip арналған көптеген танымал бағдарламалық пакеттер болады. Linux-та, сонымен қатар, tar атауы бар ескі мұрағаттау тәсілі бар. Tar (мұрағат лентасы) бағдарламасы бастапқы бетте түйін орнына жиналған файлдар жинағы және ақпаратты ленталы тасымалдаушыда сақтау үшін пайдаланылған. Ал бүгінгі таңда, tar ең алдымен файлдарды мұрағаттауды Интернет арқылы берілуінде пайдаланылады.

Tar бағдарламасы сығымдамайды, сондықтан мұрағатты файлдың көлемін кішірейту үшін, оны сығымдау керек.

Жоғалтулы сығымдау негізінен суреттер, дыбыс және бейне файлдарында қолданылады, ал жоғалтусыз сығымдауда бағдарламалық жасақтаманың мәтіндік және бастапқы файлдары қолданылады. Сығымдау ауқымы, яғни, файлдың көлемін кішірейту қолданылған сығымдау алгоритмінен және деректердің өзіне байланысты болады. Мәтіндік файлдарды жиі сығымдау арқылы 88%-ға кішірейтуге болады. Аудиоларды жоғалтулармен сығымдау әдетте файл деректерінің бастапқы нұскасынан 10%-ға дейін немесе 20 %-ға дейін сығымдауы мүмкін, ал аудио файлдарды жоғалтусыз сығымдау файлдардың бастапқы көлемінен 50%-ға дейін кішірейтуі мүмкін.

Файлдар, Тізімдемелер және файлдық жүйе

131

Файлдық жүйелер және дисктер

Файлдық жүйе – физикалық және логикалық файл кеңістігін қамтитын иерархиялық құрылым. Файлдық жүйе белгі индексіне арналған логикалық файлдың атауынан физикалық орналасуға дейін қамтылады. Файлдық жүйеге қосылу және өшірілу кеңістігі арқылы файлдық кеңістіктің өсуі немесе қысқартылуы кіреді. Бұны кейде *монтаж* деп атайды.

Файлдық кеңістіктің қиыстырылуы – бұл кейіннен өшірілуі мүмкін. Файлдық жүйе екінші деректерді сақтау құрылғысынан артықшылығы көбірек ішкі қатты дисктан сыртқы оптикалық жетектен және қатты дисктағы сыртқы жинақтан тұрады. Бұдан басқа, пайдаланушылар магниттік лентада флеш-диск пен жинақтарды қоса алады. Желілік компьютерлер үшін файлдық кеңістік өзіне жойылған дисктер кеңістігін кіргізе алады (жиі файлдық сервер деп аталады).

Операциялық жүйе файлдық кеңістікті екі деңгейде қолдайды: логикалық және физикалық. Файлдық кеңістіктегі біздің көрінісіміз – бұл логикалық көрініс – бөлімдер, тізімдемелер және файлдар. Файлға қол жеткізу үшін, операциялық жүйе логикалық орналастыру жолынан физикалық тұрған жеріне белгі қояды. Бұл тізімдемеде ақпарат файлдың орналасуын физикалық жазу арқылы жүзеге асыралады. Орын әдетте, көрсеткіш арқылы көрсетіледі. Мәселен, біз тек қана қатты дискте сақталатын файлдармен жұмыс жасаймыз деп болжайық. Көрсеткіш файлдық жүйенің нақты *блогін* көрсетеді. Ол бүтін саннан жетек блогына, беткейге (егер біз көп қатпарлы дискпен жұмыс жасасақ), секторға, сонымен қоса, диск жолағына ауысуы тиіс (5.3 суреті). Цилиндр термині айналмалы диск тілімшесінің барлық жолақ пен сектор нөмірлерін білдіру үшін пайдаланылады. Біз әрбір беткейде бір уақытта секторға және жолаққа мүмкіндік алу үшін цилиндрға сүйене алуымызда мүмкін. Мысал ретінде, fl.txt файлы 381551 блогында орналасқан деп елестетейік. Біздің файлдық жүйе 4 тілімшелі немесе 8 беткейлі бір қатты дисктан тұрады. Бұдан басқа әрбір беткейде 64 сектор және 2048 жолақ болады. Біз бұл блогті $381551/(64*2048)=\text{беткей } 2$ (0-дік беткейінде септемені бастау) беткейінде тауып алғымыз келеді. Беткей 2 шын мәнінде 262144-тен 393216 дейінгі блогтарды қамтиды. Біз блог 381551 блогін 58 секторында таба аламыз (тағыда 0-дік сектірінде септемені бастау). Бұл дегеніміз $(381551 - 262144)/2048=58.3$.

Сурет 5.3 Қатты диск үлгісі. (LionKimbrow жариялы иелік суретінен жартылай бейімделген, http://commons.wikimedia.org/wiki/File:Cylinder_Head_Sector.svg)

Ақыр аяғында біз 623 блогін табар едік $(381551 - 262144) - 2048 * 58 = 623$. Сонымен, 381551 блогында орналасқан fl.txt файл. 2 беткейде, 58 секторында, 623 трегінде болады. 5.3 суретінде көрсетілгендей қатты диск бірнеше тілімнен тұрады. Тілімшелер унисонды айналады, өйткені мо-

торлы айналдырғы әрбір тілімшенің саңылауы арқылы байланысты. Дисктің айналу жылдамдығы дисктің түріне байланысты, бірақ қатты диск жетегі 5400-ден 15000 дейінгі айн/мин. жылдамдығымен айналады. Әрбір тілімшеде оған байланысты, 2 оқу/жазба бастиегі болады (сондықтан тілімшенің екі беткейінде қолдануға болады). Қатты дисктің барлық оқу/жазба бастиектері унисонды ауысып отырады. Оқу/жазба бастиегінің тетігі орындаушы механизмі арқылы басқарылады, диск айналдырғының көмегімен айналып жатқан уақытта механизм оларды диск беткейі бойынша ауыстырып отырады. Іс жүзінде, оқу/жазба бастиектері диск беткейіне жанаспайды, ауалы жастықша беткейінде төмен/жоғары шарықтап тұрады, сол арқылы диск айналымының жылдамдығын тездетеді. Оқу/жазба бастиегі диск беткейіне жанаспауы маңызды, өйткені олар сонда сақталған деректерді жойып және беткейді (тырнап алу немесе мыжылу) зақымдауы мүмкін. Осы себептер бойынша дискті зақымдап алу тәуекеліне бармау үшін оқу/жазба бастиегінің тетігін компьютерге ауыстырылмас бұрын «тұрақтап» қойған маңызды. Оқу/жазба бастиегінің тетігі компьютерді сөндірген кезде тұрақталады. Көптеген замануи портативті компьютерлер оқыс қимылды білу үшін қимыл датчиктерін пайдаланады, сондықтан оқу/жазба бастиегінің тетігі жұлқу немесе ноутбук құлап қалған жағдайда тез тұрақтала қалуы мүмкін.

Деректер қатты дискте оң және теріс магнитті зарядтар түрінде сақталады. Оқу/жазба бастиегі зарядты беткейден оқуы немесе жаңа заряд жазуы мүмкін (тиісінше жүктеу/ашу және есте сақтау/сақтау әрекеттері). Оқу/жазба бастиегі тетігіне арналған сегіз тілімше (он алты бастиек, тек қана жоғары бастиегі ғана көрінген) 5.4 суретінде көрсетілген. Файлдар барлық диск беткейі бойынша дәйекті түрде сақталмайды. Оның орнына файлдар белгілі көлемді бөлімшелерге бөлінеді, олар блогтар деп аталады, ал файлдарың блогтары диск беткейі бойынша, диск беткейі, мүмкін бірнеше беткей арқылы орналасқан. 5.5 суретінде алты блогтан (0-5 дейінгі нөмірлермен) құрылған файлдың әртүрлі жерде, барлық қатты диск бойында орналасқан идеяны суреттейді. Әрбір блог бірдей көлемге ие болғандықтан, соңғы блог қатты толмауы керек, соған назар аударыңдар. Пунктерлі сызық 683 дисктік блогында үзіндіні көрсетеді, бұл дегеніміз файлдың соңы барлық диск блогын толтырмайды, өйткені оның кейбір бөліктері қолданылмайды.

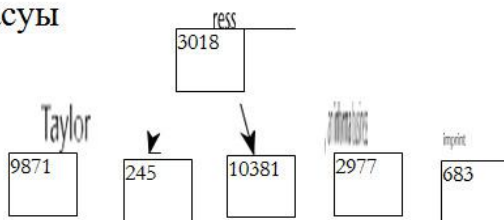


5.4 суреті Оқу/жазба бастиегінің тетігі. (Hubert Berberich көрсетілген, <http://commons.wikimedia.org/wiki/File:Seagate-ST4702N-03.jpg>.)

Файл

блок	блок	блок	блок	блок	блок
0	1	2	3	4	5

Диск блогының орналасуы



5.5 суреті Файл блоктарының диск блоктарында белгіленуі

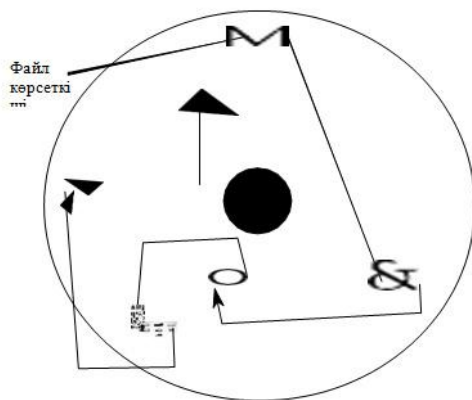
Файлдық блогтарды барлық диск беткейлерінде шашыратудың бірнеше себептері болады. Біріншіден, диск айналымының жылдамдығынан, егер оқу/жазба бастиегі бірнеше дәйекті түрдегі диск блоктарынан мүмкіншілік алуы тиіс, блок ішіндегі бағалыны жадқа жіберу үшін көп уақыт кететіндігіне қарағанда, ішкі оқу/жазба бастиегінің тетігін келесі блокқа ауыстыру үшін дискке соншалықты уақыт керек. Оқу/жазба бастиегі келесі қол жеткізу үшін позициядан тыс болуы керек, демек, қол жеткізу, яғни, басқа диск айналымынан оқу/жазба бастиегінің блогы айналғанша күтілуі тиіс, осы нәтиже болып табылады. Бұл мәселемен күресудің бір шешімі дисктік блоктарды енгізудің жалпы орны болады, бірақ шектеспейді. Мысалы, диск блогы жолақтың кез-келген секторында, содан кейін шектеспейтін жолақтарда сақталуы мүмкін (

Мысалы, 53 жолақ/ 0 секторы, 53 жолақ/ 2 секторы, 53 жолақ/4секторы, 53 жолақ/6 сектор, 53 жолақ/ 8 секторы, 54 жолақ/ 0 секторы, 54 жолақ/ 2 секторы).

Екіншіден, бұл фрагментуеу мәселесін жояды. 5.5 суретінде көрсетілгендей, фрагмент тек соңғы дисктің файлдық блогын құрайды. Егер дисктегі файлдар бірізді ұяшықтарға ауыстырылуы мүмкіндігін қарастырайық. Файл 3-ші бірізді блокта сақталған деп елестетейік. Содан кейін, бірден екінші файл 2-бірізді блокта сақталады. Үшінші файл екінші файлдан кейін 4-бірізді блокта сақталады. Екінші файл түзетілген және кеңейтілген. Ол 2-ші блокқа сыймағандықтан, Енді бізде 1 файлда (3 блок), босатылған орында (2 блок), 3 файлда (4 блок) және 2 файлда (3 блок) бар. Өкінішке орай, 2-ші блоктан құрылған еркін кеңістік пайдасыз болуы мүмкін, сондықтан жаңа файлдарға сақтау үшін анағұрлым көбірек орын қажет болар. Сөйтіп, бізде 2 блоктан құрылған фрагмент бар. Қайта қолдану үшін бізге кейбір сәтте 2 блоктан құрылған кеңістік қажет болар, бірақ ол біршама ұзақ уақыт аралығында пайдаланылмай қала алады. Егер сіз бейнеге немесе ұнтаспаға бірнеше нышанды жазбасаңыз, онда сізге алдыңғы нышандар керек емес деп шешуіңізге әбден ұқсас. Сізге қажет емес кеңістіктегі элементті толықтыру жағдайын істеу немесе істемуіңіз мүмкін. Бірізді дисктік блоктарды пайдалану, барлық файлдық жүйе бойынша керек емес ондаған немесе жүздеген фрагменттердің құрылуына әкеп соғуы мүмкін, сондықтан біз файлдарды жиі түзетіп және қайта сақтаймыз. Сөйтіп, блоктарды орналастыру жолымен, бізге жаңа блок керек кезде, біз кез-келген бос блокты кез-келген уақытта пайдалана аламыз.

Үшіншіден, файлдық жүйедегі бос кеңістіктің қолжетімділігін сипаттауын қолдау жеңілірек, өйткені дисктік файл блоктарын қадағалау үшін сол бір механизм қолданылады. Босатылған кеңістік мынадай дисктің блокторынан тұрады: әлі қолданылмаған немесе өшірілген файлдардан. Сөйтіп, тегін қолданылатын мониторинг және жазба , қосымша дисктің кеңістіктігі алмайды.

Бұл нақты сұрақтарға алып келеді: егер файлдық жүйеде таратылған файл дискісі блоктарға бөлінсе, онда осы блокты ОЖ қалай табады (мысалы, үшінші блок файлы)? Операциялық жүйе әрбір дисктік файлдың бірінші тізімін қолдайтынын, еске түсірейік. Әрбір диск блогы келесі дисктік блок файлында сақталу орнын бөлік ретінде құрайды. Сақтау көрсеткіш болып табылады. Бұл *байланысты тізім* деп белгілі. ОЖ бірінші блогтік файлдықкөрсеткішті сақтайды, әрбір блок келесі файл блогында көрсеткішті сақтайды.



5.6 Суреті. Көрсеткіштер келесі дисктағы файл блогын анықтау үшін қолданылады.

Операциялық жүйе бірінші файл көрсеткішін алу үшін ең алдымен тізімдеме тізімін тексереді, содан соң жеке диск блогын жүктейді. Бірінші блок екінші блогтың көрсеткішін, ал екінші үшінші блок көрсеткішінен тұрады. N блогына жету үшін, ең алдымен көрсеткішке бағына отырып алдыңғы блоктар арқылы n-1 блогынан өту керек. Байлынысқан тізім бойынша жүру тиімді емес болып табылады, өйткені әрбір блок дискқа қолжеткізуді талап етеді, жоғарыда айтып кеткендей, бұл кез-келген компьютерді баяулататын аспект болып саналады. Дәйекті түрде тұрған файл үшін, яғни, басынан аяғына дейін қол жетімді болуы тиіс, соның арқасында, файлға қол жеткізу жетіспеушілік болып табылмайды, сонда ғана, файл блоктан блокқа дейінгі тәртіппен жүктеледі.

Дисктік утилиттер

Біз 4 тарауды еске ала отырып ОЖ жарты бөлігі сервестік бағдарлама жасақтамасынан тұратынын аңғарамыз. Бұл бағдарламалық жасақтама мәселен өзіне вирусқа қарсы бағдарламаны кіргізе алады. Басқа пайдалы бағдарламалар сыныбы диск утилиттерінде. Дисктік утилиттер пайдаланушы мен жүйелік әкімшіге файлдық жүйені басқаруға және қолдап отыруға көмек көрсете алады, ал қол жетімді болмаған мәселені орындай алады. Төменде кең таралған дисктік утилиттердің тізімі берілген.

Дискіні дефрагменттеу: барлық файлдық жүйе бойынша шашылған

диск блоктарын қолдану фрагмент құрудың алдын алса да, фрагменттер бәрі бір де пайдалы болуы мүмкін. Дискіні дефрагменттеу диск блоктарын диск беткейі арқылы ауыстырады, оларды бірге ауыстыруы және қолданылатын блоктардың арасындағы бос блоктарды өшіреді. Бұл диск өнімділігін тиімді етуге тырысады, сол кезде файл блоктарын бір-біріне жақын ауыстырады, сондықтан бар назар файлға ауысқан кезде қол жетімділік барынша кішірейеді, өйткені барлық блоктар бір ортақ беткейде орналасады. Windows файлдық жүйесі осы кезде дефрагменттеуден пайда көруі мүмкін, бірақ Linux файлдық жүйесі уақыт мағынасына тиімді емес болмайды.

Файлдарды қалпына келтіру: егер сіз ең алдымен файлды өшіріп, содан кейін себетті тазалаңыз, файл кетті деп ойлайсыз, солай емес пе? Жоқ бұл олай емес. Файлдарды қалпына келтіретін бағдарлама файлды біріктіріп жинап қалпына келтіруге тырысады, дисктік блоктар файл болып тұрғанша, басқа жаңа файлға көшіріліп жазылғанынша.

Деректердің резервті көшірмесі: көптеген пайдаланушылар файлдық жүйеге резервті көшірме жасамайды, сондықтан кейде аса бір күрделі қателер туындаған кезде олардың барлық деректері жоғалады (мыңдаған деректер). Резервті көшірме жасау маңызды болып саналады, бірақ оны басқару кейде қиынға түседі, сіз барлығының немесе осының ғана резервті көшірмесін аласыз, соңғы кезде өзгерген резервті көшірмеден резервті көшірменің утилитасы сізге резервті көшірме үрдісін басқаруға көмектеседі.

Файлдар, тізімдемелер және файлдық жүйе

135

Кейбір файлдарға қол жеткізу кездейсоқ болуы мүмкін, яғни, кез-келген блокқа қол жеткізу қалаған уақытта бола алады. Осыны ескере отырып, кейбір операциялық жүйелер, Windows-ты қоса алғанда олар файлдарды орналастыру кестесін пайдаланады(FAT). FAT FAT дисктің басындағы сақталатын әрбір дисктік блогтың көрсеткіштерін бір файлға жинайды. Солан кейін файл жадыға жүктеледі, дисктің орнына, осы блоктың орналасуын табу үшін. ОЖ жадынан іздеу керек. Диск іздеуінен карағанда, блокқа арналған дисктің жадын іздеу аса тиімді болады. 5.7 суретінде FAT-ң жарты бөлігі көрсетілген. Әрбір блоктың FAT дискісында өзінің кезекті блогымен сақталатынына назар аударған жөн. Сондықтан, біз бұл жағдайда 150 блогынан тұратын файлдың келесі 381 блогындағы орнында, ал 151блогынан тұратын файл 153 орында және 153-тен кейін 156 орналасуы бұл соңғы орналасу болып саналатынын көріп отырмыз. 152 жаман сектор болып саналады, ал 154,155 және 732 басқа файлдың бөлігі болады. EOF «файлдың соңы» даму білдіреді.

Еркін кеңістікте байланысты тізім көмегімен сақталады, онда ерікн

жүйе файлдық жүйе бөлімінде арнайы көрсеткішпен көрсетіледі, содан кейін әрбір келесі еркін блок алдыңғы еркін блокпен көрсетіліп отырады. Файлды өшіру үшін, көрсеткіштің бір-екеуін өзгерсе болғаны. Соңғы еркін кеңістік көрсеткіші бірінші блокта өшірілген файлды көрсетіп тұрады және тізімдегі жазба бірінші блогтің бос мағынасына ауысып көрсетіледі(файлдың енді жоқ екенін көрсету үшін). Егер өшірілген файл блогтары қайта қолданысқа ие болмаса, өшірілген файлды (файлды қалпына келтіру) жеңіл қалпына келтіруге болатынына назар аудару керек. Осы жағдайда, өшірілген файл дисктан шартты түрде өшірілмейді, оның орнына диск файл блоктарын бос блог тізіміне қосады. Осыны алдыңғы беттегі жаяу жолда қарастырғанын көреміз.

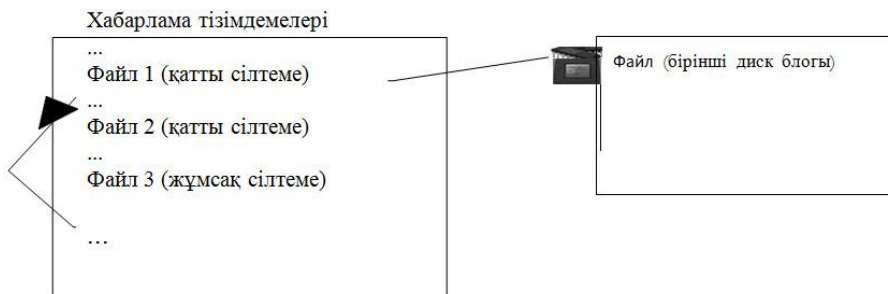
Файлдық жүйедегі(тізімдемедегі) көрсеткіштер дисктің физикалық орналасуында кейде *қатты сілтемелер* деп аталады. Яғни, файл атауларының байланысы қатты байланыс болып табылады. Linux файлда бірнеше қатты сілтемелі көрсеткіштер бола алуына мүмкіндік береді, кейбір операциялық жүйелер жұмсақ сілтемені де ұсынады. Файлдық жүйедегі көрсеткіш жұмсақ сілтемеде файлдың физикалық орнын көрсетпейді, тізімдеме құрлысының басқа жазбасын көрсетеді. Жұмсақ сілтеме Linux-та символдық сілтеме, ал Windows-та жарлық ретінде белгілі, олар түпнұсқаны енгізуде көрсетіледі. Мысалы, егер сіз /home/foxr директориядағы f1.txt файлын құрсаңыз содан кейін , /home/foxr/cit130/stuff-да f2.txt атымен жұмсақ сілтеме құрасыз, сонда /home/foxr/cit130/stuff тізімдемесі f2.txt жазбасына ие болады, содан кейін осы файлдың өзі бірінші блокқа емес /home/foxr/ f1.txt көрсеткішін көрсетеді. Мысал ретінде, ФАЙЛ1 және ФАЙЛ2 атты екі қатты сілтемені бірінші файл блогында көрсетілгенін 5.8 суретінде көре аласыз, File 3 жұмсақ сілтемесі, File 2 қатты байланысын көрсетеді.

Кез-келген байланысты пайдаланар алдында лақап аттар құрылады. Лақап ат нақты нышанға (бұл жағдайда файл) бірнеше атау көмегі арқылы сүйенуге болады. Лақап аттар қауіпті болады, өйткені біреу элементті өзгерсе, ол басқада пунктті өзгерткенін түсінбейді. Мысалы, мен f1.txt файл өзгерткін келсе, онда f2.txt файлыда өзгереді, өйткені f2.txt тек қана f1.txt көрсеткіші болып табылады. Бастапқы файл өшіріліп қалғанда жұмыс нашарлайды.

Файлдар орналасу кестеі (бөлік)

блок	150	151	152	153	154	155	156
Келесі орын	381	153	Bad	156	155	732	EOF

Сурет 5.7 файлдар орналасу кестісінің бөлігі



Сурет 5.8 Қатты сілтемелер және жұмсақ сілтемелер.

Өшіру үрдісінде тұрған файлды көрсету үшін, жұмсақ сілтеменің бар екеніг көру үшін ОЖ тексеруді қажет етпейді. Нәтижесінде файл мүлде жоғалады, бірақ жұмсақ сілтеме соны меңзеп тұрады. Бұл жағдайда, жұмсақ сілтеме бойынша жүру қате болып есептеледі. Физикалық файлды өшірмей, жұмсақ сілтемені өшіруге болады, бірақ бастапқы файлды жұмсақ сілтемесіз өшіру көптеген қателіктерге әкеліп соғады.

Процессордың жылдамдығы немесе негізгі жадыға қол жеткізуге қарағанда, дискқа қол жеткізу әлде қайда баяуырақ болады. Дискқа қол жеткізу уақыты әртүрлі факторлардың қиыстыруынан көрінеді. Мекен-жай логикалық бағытан диск блогына FAT келесі көрсетумен физикалық орналасуда беткей, жолақ және сектор тұрғысынан белгіленуі тиіс. Бұл өте тез орындалады, өйткені белгі дисктің орнына процессор жады қол жетімділігінің көмегімен орындалады. Ендігі кезде оқу/жазба бастиегі дисктағы тиісті орынға орналастырылуы керек. Бұл оқу/жазба бастиегін дұрыс жолмен ауыстырылуы (*іздеу уақыты* сияқты белгілі) және белгілен оқу/жазбасы секторда күтілуі керек (*айналу кідірісі* немесе *айналмалы шертпе* атымен белгілі). Енді, оқу/жазба бастиегі дисктағы магнитті зарядтың оқылуынан және жазылуынан басталуы мүмкін. Бұл әрекет көп уақытты талап етеді, бұл демек қол жеткізу DRAM (динамикалық RAM) арқылы орындалады. Биттер жазылып және оқылып отырылған соң, онда ақпарат диск пен жады арасына берілу керек. Бұл *берілу уақыты* деген атаумен танымал. Белгіні үйлестірудің іздеу уақыты, айналу және уақыт кідірісі бірнеше мильді секунд тәртібінде берілуден тұрады (5-тен 50мс дейін мүмкін). Миллиардты секунд мыңдық секунд жартысы. Біздің процессор және статистикалық оперативті жады нанолы секунд және (SRAM) жылдамдығы жұмыс істейді, 25-тен 100-ге дейін баяуырақ, бірақ әлі де ондық нанолы секунд диапазонында болады. Сөйтіп, процессордың шертпесі және жадының жылдамдығына

карағанда, дискқа қол жеткізу уақыты миллион рет баяу факторы болып саналады .

Linux файлдық жүйесі

Linux файлдық кеңістігі

Linux файлдық кеңістігі тізімдеме тамырының деңгейі әдепкі қалпы бойынша орнығалы, сонда Linux орнату кезінде негізгі файлдармен толығады. Бұл белгі файл түрлерін қайда тұрғанын табуға көмектеседі. Бұлар стандартты тізімдемелер:

- /bin: ОЖ ls, grep, cp, mv сияқты ортақ бағдарламалары.

- /sbin: /bin-ға ұқсас, бірақ бұнда сақталатын бағдарламалар бұл Linux командалары, осы командалар жүйелік әкімшіге арналған, бірақ бұлардың көбі кәдімгі пайдаланушыларға да қол жетімді бола алады.

Файлдар, Тізімдемелер және файлдық жүйе 137

Ескертпе: /sbin пайдаланушы жолының бөлігі болып табылады, сондықтан пайдаланушы командаға толық енгізу қажет, мысалы, /sbin/ifconfig.

- /etc: кескіндеме файлдары арнайы машина үшін (осы тарауда төменде толығырақ көрсетілген)

- /root: тамырлы пайдаланушыға арналған үй тізімдемесі. Ол әдеттегідей басқа жүйелі пайдаланушыларға тиімді емес.

- /lib: динамикалық байланыс үрдісінде қолданылатын ортақ кітапханалар.

- /dev: файлдар құрылғылары. Бұл жүйедегі әртүрлі құрылғылармен байланысуға көмектесетін арнайы файлдар. Кейбір маңызды құрылғылар төменде көрсетілген:

- /dev/fd0 – иілгіш диск (егер сондай бар болса)

- /dev/hda0 – IDE бірінші контроллерінде IDE шебер дискісі.

- /dev/ht0 – бірінші ленталы IDE тасмалдаушысы.

- /dev/lp0 – параллельді принтердің бірінші құрылғысы

- /dev/null – құрылғы емес, dev /null сүйене отырып, сіз монитор бетінде көрсеткің келмесе, сол бағдарлама шығаратын бағыттың орнына жұмсалады.

- /dev/pcd0 – CD-ROM жетегінің бірінші параллельді порты.

- /dev/pt0 – лентаның бірінші параллельді порты.

- dev/random және /dev/urandom – кездейсоқ сандардың генераторы (енді криптографиялық алгоритм үшін кездейсоқ сандар генерациясы қоолданылса, онда urandom потенциалды мәселелерге ие болады) .

- /dev/sda0 – SCSI бірінші шинасындағы SCSI бірінші дискі.

- /dev/zero – бұл көп нөлдерді алудың тәсілі.

Ескертпе: 0-ге аяқталатын құрылғы үшін, егер сізде сол түрге байланысты басқа құрылғы болса, сіз жай ғана санды улкейтуіңізге болады, мысалы, екі қосымша принтер үшін lp1, lp2 немесе CD-ROM екінші жетегі үшін pcd1.

- /tmp: бағдарламаны қосуға арналған уақытша файлдарды сақтау үшін, уақытша файлдарды қолдану және құру талап етіледі.
- /boot: бастапқы жүктемедегі қолданылатын файлдар, мысалы, lilo (Linux Loader) немесе grub (Жүктеуші Grand Unified)
- /tmp жүйелік әкімші құрған уақытша бөлімдер монтаждау нүктесін қолданады (ол әрқашан монтаждалатын файлдар болмайды).

138 Ақпараттық технологиялар

- /usr, /var, /home: басқа файлдық жүйелерге арналған монтаждау нүктесі.
- /home – пайдаланушылардың файл кеңістігі
- /var – әртүрлі бағдарламалармен сақталған деректерді орындау, журналдар файлын, электрондық пошта файлын, принтердің буферлік файлын және бұғатталған файлдарды қосқанда.
- /usr – бағдарламалық жасақтаманың әртүрлі жүйелік және пайдаланушылық қосымшалары, ішкі тізімдемемен
- /usr/bin: бұнда Linux пайдаланушылық командалары сақталады, бірақ олардың кейбірі /bin /usr/local/bin сақталады.
- /usr/sbin: бұнда жүйелік әкімшіліктің командалары сақталады.
- /usr/share/man, /usr/share/info, /usr/share/doc: құжаттардың әртүрлі беттері мен сызбалары.
- /usr/include: C бағдарламалық тіліне арналған тақырыпат файлы
- /usr/lib: бағдарламаға және жүйегі арналған өзгертпейтін файл деректері.
- /usr/local: қолданбалы бағдарламалық жасақтама және басқа файлдар.
- /proc: бұл ерекше жазба, яғни, бұл шын мәнінде файлдық жүйеде сақталатын физикалық тізімдеме емес, бірақ оның орнына ол операциялық жүйе жадында сақталады, драйвер құрылғысы тізімі сияқты, үрдіс пайдалы ақпаратты сақтайды, ол жүйе үшін бірлікпен болады, осындай тоқтату осы сәтте маңызды болып саналады, процессор жайлы ақпарат та, және сонымен қоса белсенді үрдістер.

/etc-да жүйелік әкімшіліктің көптеген маңызды файлдары болады. Осында кейбір ең маңызды файлдар төменде көрсетілген:

- қосылу сценарийі
- /etc/inittab: бастапқы қосылу сценарийі, бұл қосылу деңгейін орнатады және басқа сценарийлерді шақырады (Linux-ң анағұрлым жаңа нұсқа-

сында, бұл сценарий /etc/init бағдарламасына ауыстырылған).

- /etc/rc.d/rc0.d, /etc/rc.d/rc1.d, /etc/rc.d/rc2.d және т.б.: қызметке арналған қосылу сценарийі символдық сілметі тізімдемесіне көрсетеді, тізімдеменің әрбір тізімінде инициализация кезінде жүйені, негізгі қосылу деңгейінде қандай қызметке қосуды және қоспауды анықтайды (Пішіндерді басқару үрдісінің бөлімінде қосымша ақпарат алу үшін 4 және 11 тарауда сипатталған).

- /etc/init.d: қосылу сценарийлерінің көптігі осы тізімдемеде сақталады.

- бұл қосылу сценарийі 11 тарауда қаралады.

- пайдаланушылардың есеп жазба файлдары.

- /etc/passwd: пайдаланушылардың есеп жазба деректерінің базасы, оларға пайдаланушылар атының жиегімен, шын атымен, үй тізімдемесімен, қабықшаға енумен, және де басқа ақпараттар кіреді. Дегенмен де, бұл құпиясөз файлы болып аталса да, ол құпиясөздер онда сақталмайды, өйткені ол файлды кімде-кім оқуға рұқсат жоқ, сонымен қоса, онда құпиясөздерді орналастыру қауіпсіздік қауіне әкеп соғады; сондықтан құпиясөздер саялы файлда сақталады.

Файлдар, Тізімдер және файлдық жүйе

139

- /etc/shadow: пайдаланушылардың құпиясөздерін шифрланған күйде сақтайды.

- /etc/group: /etc/passwd ұқсас, пайдаланушылардың орнына топтарды сипаттайды.

- /etc/sudoers: пайдаланушылардың тізімі және қолжеткімділік құқығы, пайдаланушылардың қолжеткімділік құқығын әдеттегі айрықша реттерін көрсетеді.

-Бұл файлдар 6 тарауда қарастырылады.

- желі кескіндемесінің файлы

- /etc/resolv.conf: жергілікті машинаның DNS-серверін айтып шығу (сервер жүйесінің доменді аты)

- /etc/hosts: IP-мекен-жайының машинасын жиі қолданатын түйін атының тізімін сақтайды, бұл кезде DNS тізімі керек емес.

- /etc/hosts.allow, /etc/hosts.deny машина мекен-жайының тізімін сақтайды, қолжетімді журналда орналасқан немесе орналаспаған.

- /etc/sysconfig/iptables-config: Linux брандмауэрiнiң кескіндеме файлы, бұнда IP-мекен-жайы, порттар, хабарлама түрлерінің шектеуі немесе рұқсаты орнатылады.

- /etc/xinetd: Интернет конфигурациясының файлы, сервер қызметінің белгілері, мысалы, telnet на 23/TCP белгісі, бұндағы 23/TCP – бұл порт және сервер нөмірі, осы нөмірді telnet өңдейді; бұл inetd файлы конфигурациясының қауіпсіздік ауысымы.

-Бұл файлдардың кейбірі толығырақ 11 және 12 тарауда қарастырылады.

- /etc/fstab: жүйе инициализациясының құрастыру уақытында белгіленген файлдық жүйелер (тағы да mount-а командасы арқылы шақырылады).
- /etc/mtab: құрастырылған файлдық жүйенің тізімі, автоматты түрде mount және unmount командасы арқылы жаңартылады және df командасы арқылы пайдаланылады.
- /etc/mime.types: файлдар түрлерін анықтайды; басқа командаларға және файлға арналған конфигурациялық файл, бұл командалар осы файл түрін біліп түсіндіреді.

• Хабарлама файлы

- /etc/issue: жүйелік әкімшілікке арналған мазмұнын, жүйеге сәлемдесу мәтінін немесе қысқаша сипаттама құрайды,
- /etc/motd: күндік хабарлама, жүйеге кірген кезде автоматты түрде шығарылады, жүйелік әкімшіге арналған, әдетте әрбір пайдаланушыға жеткізілу үшін қолданылады, мәселен жоспарланған ескерту сияқты.

140 Ақпараттық технологиялар

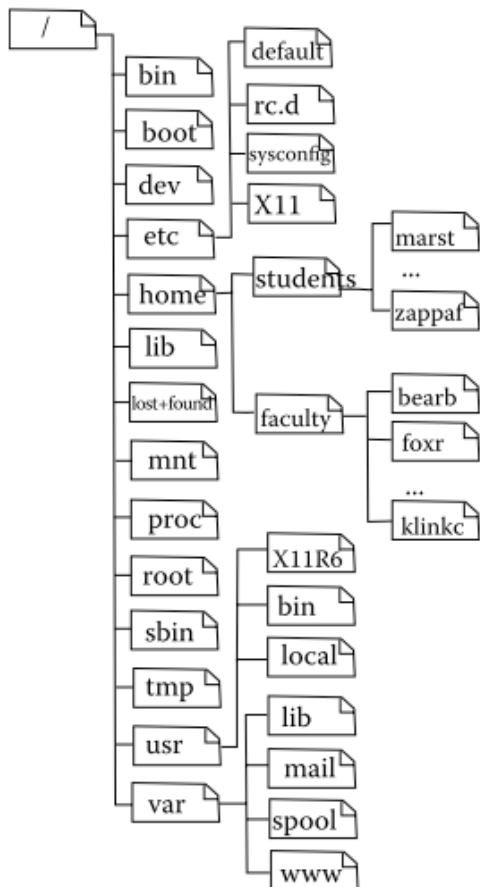
• пайдаланушылардың қосылу сценарийі

- /etc/profile, /etc/bash.rc, /etc/csh.cshrc: Bourne, BASH, немес C қабықшаларының қосылу уақыты немесе жүйеге кірген кездегі орындалатын файлдар. Бұл жүйелік әкімшіге ғаламдық мағына білдіретін әдепкі қалпы бойынша барлық пайдаланушылар үшін орнатылады. Өзінің ортасын дербестендірі үшін пайдаланушылар өздірінің үй тізімдемесінде бөлек көшірме құра алады.

-Бұл кейбір файлдар толығырақ 11 тарауда қарастырылады.

- /etc/syslog.conf: қандай жағдайлар тіркеледі және қай жерде тіркелетінін айтып жаздыратын файл конфигурациясы (біз мұны 11 тарауда толығырақ қарастырамыз)
- /etc/gdm: Gnome Display Manager (Linux GUI жүйесінің бірі) арналған файл конфигурациясы және инициализациясы.
- /etc/securetty: Қауіпсіздік терминалдың теңестіру, бұл дегеніміз, тамырлы пайдаланушыларға жүйеге кіруге рұқсат беретін терминалдар, сондықтан, тек виртуалды консольдар есептелінеді, жүйеге желі немесе модем арқылы кірген кезде кереметпайдаланушыға артықшылық алуды керек етпеу үшін.
- /etc/shells: қабықшаның сенімді тізімдері, chsh командасын пайдалана отырып, пайдаланушыларға командалық жолдан жүйеге кіру үшін қабықшаны өзгертуге мүмкіндік береді.

Дегенмен, Linux түрлендірілуі әртүрлі нақты пішінде ауысып отырады, мәселен, GUI түрінің қосылу скрипті сияқты, сонымен қоса, қызметтің түрі және атауы, олардың көбі жоғарыда көрсетілген тізімдеме және файлының Linux диалекттеріне немесе ұқсас тізімдеме файлына ие болады. 5.9 суретін қара, бұл сурет бастапқы ішкі тізімдемесінің кейбір типтік тізімдемелерінің орналасуын көрсетеді.



Сурет 5.9 Linux тізімдемесінің типтік құрылымы

/Home тізімдемесі пайдаланушылардың санына байланысты ерекшеленеді. Linux файлдық жүйесінің қалыпты құрылымы жүйелік әкімшіге Linux нұсқасын басқа Linux нұсқаларына ауысуға көмектеседі. Сөйтіп, әрбір жаңа ашылған Linux нұсқасы өз алдына өзгертулер енгізеді, сондықтан жүйелік әкімшіге сол өзгертулердің қарқынымен жүруге тура келеді.

Linux файлдық жүйесі тек қатты дисктан ғана емес, монтождалатын (алмалы-салмалы) құрылғылардан тұрады. Құрылғыға қол жеткізу үшін, ол физикалық тұрғыда қосулы болуы керек, бірақ логикалық жағынан mount командасы арқылы орнатылуы тиіс. Жүйені жүктеу уақыты кезінде кейбір монтаждаулар /etc/fstab (файлдық жүйенің кестесі) атымен монтождалынып сіздер үшін жасалады. Файлдық жүйеден бір нәрсе ағытып тастау үшін umount командасын орындаңыз. Барлығын fstab файлында өзгерту үшін mountall командасын пайдаланаңыз. Ескертпе: Егер сіз тамырда болмасаңыз, сіз fstab-ты тұжырымдай алмайсыз немесе mount, umount немесе mountall командаларын орындай алмайсыз. Тағы бір, /etc/mtab файлы қазіргі уақытта орныққан бөлімдерді кестесін сақтайды.

Linux Бөлімдері

Әртүрлі тізімдемелер айтарлық орынға ие болатынына кепілдік беру үшін, жүйелік әкімшіліктер, әдетте, қатты дискті үш немесе одан да көп салаға бөледі, олардың әрбірі файлдық жүйе бөлігінде сақталады. Бұл нақты файлдық кеңістіктің салаларын қолдануға мүмкіндік береді, олар әртүрлі мақсатқа арналатын болады, бұл ядро, тартып шығару және пайдаланушының кеңістік тізімдемесі. Бөлімдер деректерді қорғау дәрежесі қамтамасыз етеді, ол өз алдына бір файлға зақым келсе, басқа файлдарға әсер еткізбейді (зақым жасаушы дереккөзге байланысты). Тағы да, Linux-та әртүрлі бөлімдерге әртүрлі қолжетімді басқарылу элементтерін орнатуға болады. Мысалы, біреу осы бөлімнен файлдарды жазып немесе өшіріп тастамауға кепілдік беру үшін, бір бөлім тек қана оқылу үшін орнатылуы мүмкін. Сонымен қоса, дисктік үлестеме бөлімін орнатуға болады, сонда кейбір бөлімдер үлестемені сақтай алады, ал кейбірі олай істей алмайды. Өз қалауынша, бөлімдер тағыда, қосымша файл кеңістігін оңай құрастыруға және бөлшектеуге рұқсат етеді.

Тамырлы бөлім (/) Linux-тің /boot тізімдемесінде сақталады, ол жүктемелі бағдарламадан, және де тамырлы деректердің жеткілікті кеңістігінен тұрады (электрондық пошта немесе суперпайдаланушылар сияқты). Ең үлкен бөлім болып /home тізімдемесі қолданылуы мүмкін, ол барлық

тізімдемелерден және пайдаланушылардың ішкі тізімдемесінен тұрады. Сонымен қатар, /var және /usr сияқты басқа тізімдемелер жеке бөлімдер болып берілуі мүмкін, немесе олар / немесе /home бөлігі болуына да болады. Swar бөлімі виртуалды жады үшін қолданылады. Ол негізгі жадының (DRAM) қосымша ретінде қызмет етеді, сондықтан қазіргі кезде қолданылған үрдіс және деректер бөлігі ретінде, үнемі DRAM жадында тұруы керек. Бұл бөлім жеткілікті жадына қажет етеді, өйткені барлық үрдіс жасалатын кодты сақтау үшін (сонымен қоса олардың деректері). Әдетте swar DRAM-тан асатаны екі есе кіші көлемде құрылады. Swar кеңістігі де екі немесе одан да артық дисктар арасында таралуы мүмкін. Егер swar кеңістігі екі қатты диск арасына бөлінсе, ол қосымша тиімділікті қамтамасыз етуі мүмкін, сонда екі бетті де бір мезгілде өңдеуге болатын еді. (4 тараудағы тартып шығару фаайлын қараңыз). ОЖ-ні орнатып жатқан уақытта файл кеңістігінің белгісі жүзеге асады және сәйкесінше, ол аяқталысымен, бөлімдер және олардың көлемдері белгіленеді. Өкіншіне қарай, сіз орнатуды сізде пайдаланушылар болар алдына іске асырасыз, пайданушы бөлімін толықтыру үшін, сізге бөлім көлемі қандай болатыны жайлы келісілген шешім жасау керек. Егер сіз бөлім көлемін өзгерткіңіз келсе, сізге барлық операциялық жүйені қайта орнатуға тура келеді, ол дегеніміз барлық тізімдемелер мен жазбалардың өшірілуіне әкеп соғады.

142 Ақпараттық технологиялар

Сізге керек жағдайда барлық операциялық жүйе файлдарының және пайдаланушылардың есеп жазбасының/тізімдемесінің резервті көшірмесін алып, операциялық жүйені қайта орнатып, бөлімдер көлемін өзгертіп, содан кейін қайтадан операциялық жүйе файлдарын және есеп жазбасын/тізімдемесін қалпына келтіруге болар еді. Бұл көп уақытты алар еді (мүмкін бірнеше сағатты алар ма) және егер бөлімдерде түпнұсқалы бөліну жөнсіз болса ғана жасалынушы еді. Бақытына орай, Linux және Unix бөлімдерд көлемінің динамикалық өзгеруіне басқа шешім береді. Бірақ оның қолданылуы оңайлыққа соқпайды. Мысалы, егер бөлімдердің көлемін өзгертіп жатқан кезде, диск бұзылса немесе қуат көзі өшіп қалса, онда деректер мәңгілікке жойылады. Файл жүйенің сақталуына, операциялық жүйенің қайта орнатуына және файлдық жүйені қалпына келтіруге қарағанда, C басқа жағынан алғанда, әлдеқайда оңайырақ.

Файлдық жүйемен жұмыс жасайтын Linux-тің біршама командалары бар. Df командасы (файлдық жүйені көрсету) сізге барлық бөлімдер байтқа қаншалықты толыққанын және қолдануда қанша пайызға ие болатын көріге көмектеседі. Жоғарыда айтып өткендей, mount және

unmount командалары файлдық жүйені құрастыруға және бөлшектеуге рұқсат береді.

Quotaon және quotaoff командалары дисктік үлестеме белсенді немесе белсенді емес екендігін қадағалауға арналған. Жүйелік әкімші дисктік үлестемелерді пайдаланушыларға да топтарға орната алады. Егер файлдық жүйеде осында дисктік үлестеме болса, онда операциялық жүйе осы пайданушылар/топтар осы файлдық жүйенің шектеулі орнатылған үлестемелерін қолданатына кепілдік бере алады.

File командасы файл деректерінің қандай түрге жататынын анықтайды. Мысалы, ол файлдағы мәтіндік файл, тізімдеме, сурет файлы немесе Open Office құжаты екенін көрсетіп бере алады. Find командасы файлдық жүйеде файлды белгілі атаумен іздеп табумен айналысады. Linux-та find командасы пайдалы құрылым болып табылса да, ол өз алдына өте қиын құрылым болып табылады.

Fsck утилитасы (файлдық жүйенің тексерісі) файлдық жүйеде үйлеспейтін файлдарға талдау жасайды. Бұл утилитта жаман секторларды, бұзылған файлдарды таба алады, өйткені, жүйе жабылғаннан кейін, олар әлі де ашық тұрады, немесе ақпарат тасымалдайтын кейбір файлдардың қателіктерін көрсетеді (мысалы, вирус). Fsck утилитасы жаман файлдарды тауып қана қоймай, оларды қалпына келтіруге тырысады. Ол файлдық жүйеде физикалық және логикалық қателіктерді іздейді. Ол тек қана қателіктерді табумен, сол қателіктерді дұрыстау тәртібінде жұмыс істейді. Fsck бағдарламасы жүйе жүктелгеннен кейін жүктеледі, сонда ол файлдық жүйенің қауіпсіздігін және дұрыстығын қамтамасыз етеді. Егер қателік пайда болған жағдайда, жүктеме үрдісі тоқтатылады, және де fsck командасына қателікті іздеп және табылған кез-келген ақауды дұрыстатуға мүмкіндік береді.

Linux-тағы индексті дескриптор

Linux және Unix- та индексті дескриптор термині файлдық жүйенің құраушысы болып саналады. Индексті дескриптор файл болып есептелмейді, бірақ файл ақпаратын сақтайтын құрылым болып келеді. Linux-ті орнатқан кезде немесе бөлімдерді файлдық жүйеге бөлген кезде, операциялық жүйе индексті дескриптор ретін құрады (индексті дескрипторға резервтелген файлдық жүйенің жалпы кеңістігі 1% құрайды). Жүйеге арналған индексті дескриптордың белгілі бір көлемі бар екендігі назар аудару керек, бірақ жүйе әлде қашан индексті дескриптордың шектеуінен шығады деген аз ғана ықтимал бар. Индексті дескриптор бір файлда ғана ақпаратты сақтайды; Сөйтіп жүйедегі әрбір файлда бір индексті дескриптор бар. Индексті дескриптор үшін құрылған файл осы пайда-

ланушының және топ басшысының файлын, орналасуын, файл түрін және көрсеткіш сақталған физикалық файлды сақтайды, бірақ индексті дескриптор файл атауын сақтамайды және де файл болып саналмайды. Оның орнына, тізімдемедегі тізімде көрсеткіш дисктағы индексті дескриптордың атауына ие болады.

Файлдар, тізімдемелер және файлдық жүйе 143

Егер файл қатты байланысқа ие болса, онда атау да, қатты сілтеме де индексті дескрипторға көрсетіп тұрады. Егер файл жұмсақ сілтемеге ие болса, файл атауы индексті дескрипторға көрсетіп тұрады, ол кезде жұмсақ сілтеме файл атауын көрсетеді.

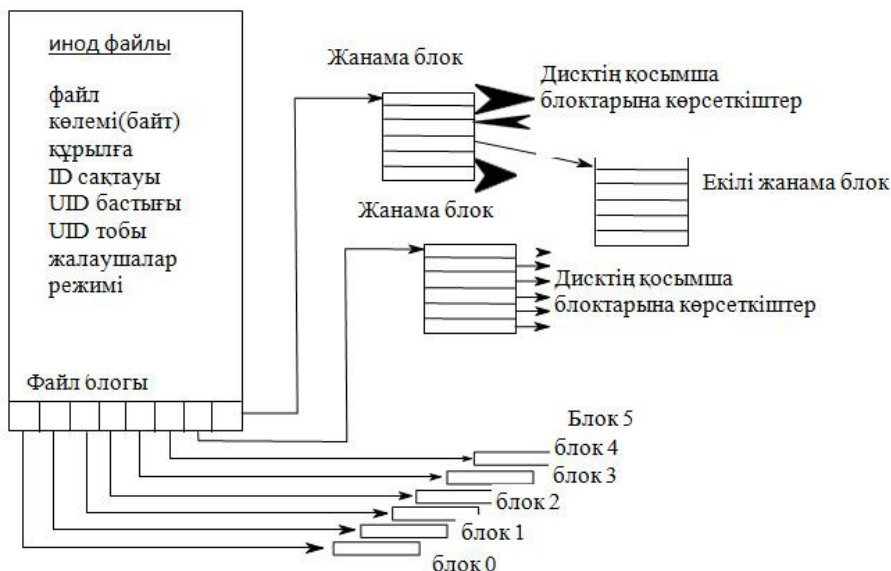
Индексті дескриптордың нөмірін алу үшін, ls (тізім) командасы –i опциясын ұсынады. Солай, ls –i файл атауы осы файлдың индексті дескрипторының нөмірін қайтарады. Сіз ақпаратты ls-l индексті дескриптор көмегімен алуыңызға болады.

Бұл тарауда айтылғандай, файлдар дисктің шектескен блогында сақталмайды, оның орнына, файлдық жүйе бойына таратылған блоктарған бөлінген. Әрбір файлға қосылу үшін бастапқы блок сандары беріледі. Индексті дескриптор файл жайлы ақпаратты сақтаудан басқа, сол бастапқы блоктарға көрсетілетін көрсеткіштер ретін құрайды. Бірінші көрсеткіш бірінші диск блогын, екіншісі көрсеткіш екінші диск блогын көрсетеді, мұнда бірінші блок дисксі файлдың бірінші бөлігін сақтайды, екінші блок дисксі файлдың екінші бөлігін сақтайды және т.б.

Егер файл қосыша блоктарды қажет етсе, кейіннен индексті дескрипторының көрсеткіші көрсеткіш блогын көрсетілгендей құралады, сонда, әрбіреуіқосымша дисктік блоктарға көрсетіп тұрады. Осындай көрсеткіш блогы *жанама блок* деп аталады. Онда кез-келген деңгей мөлшері қосыла алады, мысалы, индексті дескриптор көрсеткіші жанама блокқа көрсетіп тұрады, ол өзіне сол жанама блокқа көрсетіліп тұрады, олар шын мәнідегі дисктік блоктар көрсеткішіне ие болады. 5.10 сурет осы құрылысты суреттей отырып, диск блогының иерархиялық құрылымын, Linux-та көрсеткішін пен индексті дескрипторды көрсетеді. Индексті дескриптор Linux ОЖ FAT жүйесін қолдаудан босатылады (Windows сияқты).

Файлдардың ақпаратын сақтаумен қатар, тізімдеме ақпаратын және символдық сілтемелерді сақтау үшін индексті дескрипторді қолданады. Символдық сілтеме файлға қол жеткізу үшін баламалы жолды болып келеді. Егер сіз осы файлға қол жеткізу үшін бірнеше бастапқы нүктелерді көрсететін мүмкіндікке ие болағыңыз келсе, сіз үнемі символдық сілтемелер құрасыз. Символдық сілтеме құру үшін ls – файл_атауы1 файл_атауы2 командасын қолданыңыз. Мұндағы файл_атауы1 болған

файл болып және файл_атауы2 символдық сілтеме болып келеді. Әдеттегідей, осы екі файл атауының біреуі екі бөлек тізімдемеде файл және символдық сілтеме ретінде болады (сіз сүйеніп отырған файлдың сол тізімдемеде символдық сілтемесі болуының аса мәні жоқ).



Сурет 5.10 Linux-ғы индексті дескриптордың иерархиялық құрылымы
144

Ақпараттық технологиялар

Егер foo атты /usr/share/progs тізімдемесінде маңызды орындалатын бағдарлама бар болса, және де сіз оған үй директориясында қол жеткіңіз келсе, ln -s /usr/shar/progs/foo foo командасын орындауыңыз керек. s /usr/shar/progs/foo командасына сүйенетін, сідің үй тізімдемеңізде foo атты символдық сілтеме құрылады.

Символдық сілтемелермен жұмыс істегенде тиянақты болуы керек, өйткені кімде-кім бастапқы файлды өшіріпi немесе ауыстырса, онда сілтеме іске аспайды. Егер foo файлы /usr/share/progs-тан өшірілген болса, содан кейін ~ foo енгізу қателікке апарып әкеледі.

Компьютерлық вирустар

Кез-келген файлдық жүйені компьютерлік вирустарынсыз елестету немесе олардың қауіпсіздігін айтпай кету толық болмас еді. Компьютерлік вирустар өзін көшіріп ала алатын бағдарламаны көрсетеді. Компьютерлік вирустардың басты мақсаты компьютерді зақымдау болып табыла-

ды, ол сол компьютерде сақталады және сол компьютердің пайдаланушысынан қалмай андиды.

Оны қалай жұқтырып алуға болады? Вирус басқа орындалатын файлда тығылып отырады. Осы бағдарлама іске асқан кезде, вирус коды іске асады. Вирус коды әдеттегідей, екі бөліктен тұрады, біріншісі өзіні көшіреді және өзінің орындалатын кодын файлдар деректінде немесе басқа бағдарлама ішінде сақтайды. Екіншісі зиянды бөлігі болып келеді.

Әдетте вирус зиянды бөлік орындалып жатқан белгелі бір шарттың (анағұрлым ескі вирустар белгілі бір күнді күтуі мүмкін, мәселен жұманың 13-ін, немес уақытты, әрекетті, пайдаланушы файлды өшіру тырысқан кезді) пайда болуын күтеді. Зиянды бөлік деректерді өшіріп немесе оны көшіріп алып, үшіншы жаққа жіберуі мүмкін (бұл деректер cookies-та сақталған, браузердің тарихи, немесе кредиттік карта нөмірлерінің ақпараттарын қамтуы мүмкін).

Бүгінде «вирус» сөзі көптеген әртүрлі зиянды бағдарламалық жасақтамаларға жатады олар: тыңшы-бағдарламалар, жарнамалы БЖ, троянды аттар, құрттар, рут-киттер. Дегенмен, вирус термині тек қана өзін-өзі орындайтын бағдарламалық жасақтамаға жатуы мүмкін, олар өздеріне тыңшы-бағдарламаны, жарнамалы БЖ немесе трояндық аттарды қамтымайды. Біріктірілген бағдарламалық жасақтама тобы енді зиянды бағдарлама деп аталады.

Одан қалай қорғануға болады? Интернеттен арылу ең басты амал болып табылады (электрондық поштаны қоса алғанда)! Бірақ бұл пайдасыз. Оның орнына соңғы маңызды компьютерді вирусқа қарсы бағдарламалардың болуы аса мәнді болып табылады. Тағы да cookies ағытып тастау көмектеседі. Одан басқа, өзіңіз білмейтін адамдардың әрекетін ашпаңыз. Тағы да бірнеше апаттық қалпына келтіру жоспарларына ие болған пайдалы. Жаман жұқтыру жағдайында – сізге файлдық жүйенің резервті көшірмесін алған дұрыс, сол кезде сіз операциялық жүйені қайта құрып, файлдарды қалпына келтіре аласыз. Егер сіз Linux немесе Mac пайдаланушысы болсаңыз онда зиянды бағдарламалардан қауіпсізбін деп санауыңызға болады. Көптеген зиянды мақсатты бағдарламалар Windows машиналары болып табылады, Windows кең таралған ОЖ болып есептеледі, бірақ шын мәнінде зиянды бағдарламалар кез-келген жүйеде бола алады.

In командасының -s параметрінсіз құрылған қатты байланыс, индексті дескриптор файлының көрсеткіші болып келеді, тізімдемедегі файл атауын жай ғана көрсететін символдық сілтемеге қарағанда, тізімдемедегі бастапқы атау индексті дескриптор файлына көрсетеді. Қатты сілтеме

сол файлдық жүйедегі сілтеме файлында ғана қолданылуы мүмкін. Файлдық жүйе (бөлімдер) арқылы құрастырылған сілтеме символдық сілтеме болуы тиіс. Бұл қатты сілтеме пайдалылығын шектейді. Қатты сілтемелер кең таралпаған, ал символдық сілтеме көп таралған. Егер сіз ls -l деп жасасаңыз, онда 'l'- ден тізілген файл түрлерінің символдық сілтемесін көресіз (бірінші әріпті тізу), онда файл атауының нақты орны жұмсақ сілтемемен көрсетіледі. Файл атауы – бұл сіз сілтеме құрған кезде атау, сілтеме көрсеткен нақты орын осы кезде файлға және файл атауын апаратын жол болып табылады.

Файлдар, тізімдемелер және файлдық жүйе

145

Windows файлдық жүйесі

Linux-қа қарағанда, Windows файлдық жүйесінің таралуы оңайырақ, сіз бір нәрсені іздеуге тиіс болсаңыз, ол оны қиындата түседі, өйткені, Linux-та тапқан файлдарыңызға қарағанда, сіз бір тізімдеменің ішінен әлдеқайда көп файлдар таба аласыз. Windows 95 нұсқасынан бастап келе, Windows-тың таралуы бір қалыпты болды. C: бөлімі Linux-ғы / тамырлы директориясымен баламалы. A: және B: атаулары иілгіш диск үшін резевтелінге (соңғы жылдары алырған компьютерлерде кейбіріне қол жетімділік жоқ), және D: компьютердің оптикалық жетегіне иеленеді. Басқа да бөлінулер қосылуы мүмкін (немесе негізгі қатты диск өзінің атаулы әріпі бар бірнеше «дисктарға» бөлінуі мүмкін).

(C:) тамырлы файлдық жүйеде, файлдық жүйе кем дегенде үш тізімдемеге бөлінген. Операциялық жүйе Windows атты папкада сақталады. Енгізілген папкадағы System32 тізімдемесі Linux /usr/lib ұқсас жүйелік кітапхана мен ортақ файлдардан тұрады. Жанама бағдарламалық жасақтаманың үлкен бөлігі Program Files болады. Windows 7 –де ескі 32-биттік бағдарлама жасақтамадан (Program Files(x86), 64-биттік бағдарлама жасақтамасын бөлуге арналған C:\Program Files(x86) жеке тізімдемесі бар. Linux-та жанама бағдарламалық жасақтаманың үлкен бөлігі /usr болады. Пайдаланушылардың тізімдемесі әдеттегідей, C:\Users сақталады. (Linux-та бұл тізімдемелер /home болады). Онда C:\Temp тізімдемесі болуы мүмкін, ол Linux-ғы /tmp тізімдемесіне ұқсас. Windows пен Linux-та көптеген ауытқулар пайдаланушылар тізімдемесінде болады. C:\Users-та әртүрлі файлдар түріне арналған кәдімгі қалыптағы папкалар құрылады, олар: Жұмыс үстелі, Жүктемелер, Таңдамалы, Сілтеме, Менің құжаттарым, Менің музыкам, Менің суреттерім, Менің бейнем, Іздеулер. Linux-та көбінесе элементтерді қайда орналастыратын пайдаланушы өзі шешеді. Windows операциялық жүйесінде

накты папкалар бағдарламалық жасақтамамен орнатылады, егер олар пайдаланушымен қайта жазылынбаса. Тағы да, жұмсақ сілтемелер болады, демек, «Менің құжаттарым» C:\Users Пайдаланушы аты \ Менің құжаттарым бөлігіне жатады. Файлдық кеңістікті түсіну үшін, кәдімгі қалыптағы бұл тізімдемелер және жұмсақ сілтемелер пайдаланушыға ненің қайда тұрғанын, қай файлдың қайда сақталғанын есте сақтауынан қорғап көмектеседі.

Файлдық жүйемен жасалатын көптеген Windows операциялары GUI бағдарламасы арқылы қол жетімді. Олардың көбі жаяужол басқарылуы көмегімен табылады, басқалары Norton Utilities және McAfee құралдары сияқты операциялық жүйенің утилиттары арқылы қол жетімді. Жаяужол басқарылуы өзіне жүйені қайта қалпына келтіретін, резервті көшірімені және файлдық жүйені қалпына келтіретін құралдарды кіргізеді (ол ең бірінші кезде жүйенің баптауларын қалыпқа келтіреді, бірақ файлдық жүйе компоненттерінде қалпына келтіре алады), соынмен қоса, папка ішіндегісін көрген кезде көрсетілетін папкалар опциясы. Және де, әрине Бастаушы Windows әдепкі қалпына келтірілген файлдық менеджер бағдарламасы. Тағы да ДОЖ қол жетімді командалары да бар. ДОЖ командасының ең маңыздысы chkdsk болып табылады, Linux-ғы fsck утилитасы сол мақсатта жұмсалады. Басқа командалар дефрагментеумен айналысады (диск утилитасын дефрагментеу), diskpart дискті орнықтыруға немесе дисктағы сәйкесінше әкімшілік қызметтерді орындауға рұқсат етеді (мысалы, анықтаушыға немесе дискқа әріп береді), және find Linux-ғы балама find бағдарламасы.

Windows файлдық жүйесін қарастырғанда, оның қолданылуда интуитивті жеңіл. Басқа жағынан алып қарағанда, Windows амалы тым шектеулі мәнде болуы мүмкін. Мысалы, пайдаланушы әртүрлі файлдар деректерін жинақтау үшін сол бағытқа байланысты, ол C:\130 папкасын құрғысы келеді, ал Linux-ғы пайдаланушы /home/foxr/cit130 құра алады. Осы тізімдемемен пайдаланушы барлық файлдарды бір сопақ кеңістікке орналастыруы немесе ішкі тізімдеме құруы мүмкін, мысалы, Суреттер, Құжаттар, Іздеу және Бейне.

146 Ақпараттық технологиялар

Windows операциялық жүйесінде, пайдаланушы C:\130 қатысты, барлық файлдарды жинақтыру үшін әрбір орнатылған папкалардың жанына cit 130 құруы тиіс, бірақ материалдар папкалар арасында таралмайды. Егер пайдаланушы ақылды болса, әдепкі қалыптағы мағынаны қайта анықтай алса, онда пайдаланушы барлық элементтер орныға алатын cit 130 папкасын орната алады.

Файл жүйесі бойынша жылжу

Біз мұнда Linux және ДОЖ нақты командаларының файлдық жүйеде жылжуын көреміз. Келесі мысалдарда, біз ОЖ командаларын байқаймыз, осы командалардың және кейбір пікірлер жасалған операцияның нәтижесін суреттейді. Біз ең алдымен Linux-қа көңіл бұрамыз, өйткені ол кішкене қиындау болып келеді. Linux-ті меңгере отырып, ДОЖ-ны меңгеру жеңілірек болады. Мәселен, біз 5.11 суретінде тізімдемелер, ішкі тізімдемелер құрылымына ие боламыз деп ойлайық. Тікүшбұрыштағы элементтер файл болып табылады, ал қалғандарының бәрі тізімдеме болады. ДОЖ үшін `home users` командасына өзгеріледі.

Linux

Төменде Linux сеансы көрсетіледі (Linux қабықшасында енгізілген командалар және нәтижелер), және команда түсірдірмесі. Кейбір командалар үшін пікір төменде берілген. Linux командасы `$` символынан кейін шығады, біз мұны Linux шақырылуы деп есептейміз.

```
Команда / Нәтиже
$ pwd
/home/foxr
$ cd CIT130
$ ls
HW LABS
```

```
Түсіндірулу
Жұмыс тізімдемесін басып шығару
CIT130 ішкі тізімдемесіне өту
Ішіндегісін тізіп шығу
```

Бұл тізімдемеде басылып шығатын екі пункт болады

```
$ ls -l
```

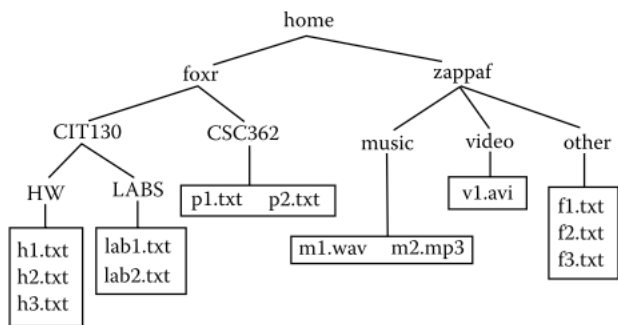
```
Ішіндегі ұзақ тізімді орындап шығу
```

```
drwxr-xr-x 2 foxr foxr 1024 Jan 20 03:41 HW
drwxr-xr-x 4 foxr foxr 4096 Jan 21 17:22 LABS
```

Ұзақ тізім анағұрлым толығырақ ақпаратты қамтиды, оған күн және құрылған уақыт кіреді (әскери уақытта, 17:22 орнына 5:22 p.m). Дәл осы сәтте біз тек бірінші элементке жүгінеміз, ол кездейсоқ көрінетін бірізділік символдары болып келеді. Бастапқы `'d'` кейбір тізімдеменің элемент екенін көрсетіп тұрады. Қалған әріптер мен дефистар құқық қолжетімділігін білдіреді (рұқсат етілу). Біз бұл тақырыпты 6 тарауда қарастырамыз.

```
$ cd hw
$ cp1.txt /home/foxr/CIT130/LABS
```

```
hw тізімдемесіне өту
LABS тізімдемесіне h1.txt файлын көшіру
```



Сурет 5.11 Бұл бөлімге арналған тізбек мысалдары

Бұл команда абсолютті жолды қолданады, LABS тізімдемесін көрсету үшін, біз тамырлы Linux деңгейінен бастаймыз.

CD ..	бір жоғары деңгейге өту (/home/foxr/CIT130)
\$ LABS mv/lab1.txt	lab1.txt мында жылжуы

. «бұл тізімдемені» білдіреді. LABS сипаттізім ішкі тізімдемесі салыстырмалы жол сияқты орындалған, ол осы нүктеден басталады.

\$ ls ../../CSC362	тізімдеме ішіндегі тізім CSC362

P1.txt p2.txt

../../CSC362 - осы сәттен бастап салыстырмалы жол, бірақ екі жоғары деңгейге жылжу (../../) және бір деңгей төмен (CSC362).

\$ cd ~/ CSC362	сіздің CSC362 тізімдемеңізге өту
\$ rm*.*	бұндағының бәрін өшіру

~ символы ағымдағы пайдаланушының үй тізімдемесін көрсетеді. Сондықтан, бұл жағдайда ~ /home/foxr/ дегенді білдіреді. Rm. -де қолданылатын *, алмастырмалы таңба болып келеді. Алмастырмалы * «еш-

теңе» білдіреді. *. * қолданылатын символ «бәрін мезгіл құрайды». Осы мезгілдегі барлық атауларды бұл команда кейіннен бәрін өшіреді (p1.txt сияқты кеңейтілген файл, ал foo сияқты таңдалмайды, өйткені мұнда мезгіл жоқ).

Rm командасы rm-i орындау үшін орнатылған, бұл дегеніміз «интерактивті тәртіп» дегенді білдіреді. Ол әрбір тізім пунктін жүргізеді, және сізден осыларды бір реттен өшіру керек пе деп, соған сенімдісіз бе сұрайды, мысалы

```
rm: әдеттегі файлды өшіру керек пе file 'p1.txt'? y
rm: әдеттегі файлды өшіру керек пе file 'p2.txt'? y
rm: удалить әдеттегі файлды өшіру керек пе 'lab1.txt'? n
148 Акпараттық технологиялар
```

Пайдаланушымен берілген жауаптар, бұл жағдайда p1.txt и p2.txt өшіруге әкеледі, бірақ lab1.txt емес (CIT130/LABS тізімдемесінен осында ауысып келген)

\$ cd ..	foxr дейін жоғары ауыстыру
\$ rmdir CSC362	CSC362 тізімдемесін толық өшіру
rmdir: CSC362 : Directory not empty	Тізімдеме бос емес

Тізімдеме бос болса, ол өшіріле алмайды. Біз тізімдемені өшіре алатын едік, егер қалған lab1.txt элементін ең басында өшіріп тастасақ.

Басқа жағынан біз rm -r CSC362 командасын орындай алуымыз мүмкін. –r операциясы «рекурсивті өшірілу» дегенді білдіреді. Рекурсивті өшірілу тізімдеме ішіндегі тізімдемені рекурсивті өшіргеннен кейін өшіріледі (егер тізімдеме ішкі тізімдемеге ие болса, онда ең алдымен ішкі тізімдеме ішіндегі өшіріледі, содан кейін тізімдеме және ішкі тізімдеме өшіріледі). Мысалы, егер біз rm -r ~foxr/CIT130 жіберсек, онда өшірі қимасы келесі ретпен орындалады: h1.txt, h2.txt, h3.txt, hw (тізімдеме), h1.txt, lab2.txt (ертерек hw тізімдемесінен h1.txt көшіріліп алынған болатын, және біз мұнда басқа тізімдемеге lab1.txt файлды ауыстырғанын еске түсірте кетейік), LABS (тізімдеме), және де ақырға, тізімдеме CIT130.

–r операциясы өте мықты, бірақ өте қауіпті болуы мүмкін. Сондықтан rm -ir. жасамасаңыз, rm –r ешқашан қолданбаңыз жөн. –i операциясы бөлігі өшіру алдында қандай бір құптауды іздеуді талап етеді. Егер сіз істеп жатқанызға сенімді болсаңыз, жүйелік әкімшіге rm –r ешқашан қолданбаған маңызды. Сіз ең алдымен cd/ орындап (файлдық жүйенің жоғарғы бөлігіне өту), содан кейін rm-r*.* орындайсыз елестетейік. Сіз

барлығын өшіргіңіз келед (сонымен қоса `rm` командасын)

Ескертпе: `cp` да рекурсивті тәртіпке `cp -r` ие болады, демек ол тізімдеме іші тізімдемені рекурсивті көшіріліп алу, `mv` рекурсивті тәртіпке ие бола алмады.

<code>\$ cd/</code>	Файлдық жүйенің жоғарғы деңгейіне өту
<code>\$ cd home</code>	Үй директорисына төмен қарай жылжу
<code>\$ ls</code>	Тізім ішіндегісін тізіп шығу
<code>Foxt zappaf</code>	
<code>\$ cd zappaf</code>	zappaf тізімдемесіне өту
<code>\$ cp music/*.*</code>	файлды музыкадан сіздің тізімдеңізге көшіру

. тағы да «барі мезгілмен» дегенді білдіреді. Осыған байланысты, бұл команда `zappaf` тізімдемесінде музыкалық файлды көшіреді, бұл үй тізімдемесін білдіретін `~` атау мезгіліне ие.

<code>\$ mv videos/*.*~</code>	бәрін ауыстыру
<code>mv: '/home/</code>	
<code>zappaf/videos/v .avi</code>	
<code>'./v.avi':</code>	

Қол жеткізуге тыйым салынады

Файлдар, тізімдемелер және Файлдық жүйе 149

Сіз `zappaf` тізімдемесіне және оның ішіндегісіне қол жеткізу үшін рұқсат ала алмасаңыз, онда сізге жылжуға рұқсат жоқ, өйткені осы пайдаланушы тізімдемесінен барлық файлдарды жылжу арқылы өшіріледі. Осы мәселені 6 тарауда зерттейміз.

<code>\$ cd</code>	өз үй тізімдемесіне қайтып келу
<code>\$ mkdir STUFF</code>	STUFF атты тізімдеме құру
<code>\$ ln -s video/v 1 .avi</code>	символдық сілтеме құру
<code>STUFF/ video</code>	

Символдық сілтеме `STUFF` ішкі тізімдемесінде сақталады (`/home/foxt/STUFF`), `b` сілтемесі бейне деп аталады. Сілтеме көрсетеді немесе сүйенеді `/home/zappaf/videos/v1.avi`.

<code>\$ ls STUFF</code>	STUFF тізімдемесі ішіндегі тізім
<code>video</code>	

`Ls` командасы ешқандайсыз бөлшексіз, жай ғана элемент атауын ғана тізіп шығады. Егер біз толығырақ білгіміз келсе, онда `ls-l` опциясын

пайдалануымыз керек («ұзын тізім» үшін).

\$ls-l STUFF

lrwxrwxrwx 1 foxr foxr 16 Jan 21 17:13 video->/home/zappaf/videos/v1.avi

Ұзын тізімнен біз бейненің орын сілтемесі /home/zappaf/videos/v1.avi. екенін көреміз. Жолақ басындағы «l» символы файлдың (сілтеменің) түрі болады және -> оңы сілтеменің қайда көрсетіп тұрғанын көрсетеді.

ДОЖ

Біз ДОЖ командасына көзқарасымызды шектеулі етеміз, өйткені олар ұқсас және мейлінше аз нұсқалы болады. Біз мұнда тек алдыңғы мысалға сүйене өтеміз.

Команда/Нәтиже

Түсіндірілу

C:

C: дискісіне/бөліміне ауысу

C: \> CD Users\foxr

foxr тізімдемесіне өту

C:\Users\foxr>dir

тізімдеме ішіндегі тізім

Тізімдемелердің ауысу кезінде, сіздің шақыруларыңыз өзгеріп отыратына назар аудару керек. Ағымдағы тізімдеме тізімінде ішінде Linux-ғы pwd эквиваленті жоқ. Dir командасы тізімдеме ішіндесін көрсетеді. Ол төменде шығарылған. Кейбір толығулар түсіріліп қалған.

Wed 11/23/2011 10:53 AM <DIR> .

Wed 11/23/2011 10:53 AM <DIR> ..

Wed 11/23/2011 10:55 AM <DIR> Desktop

150

Аппараттық технологиялар

Wed 11/23/2011 10:55 AM <DIR> Documents

Wed 11/23/2011 10:55 AM <DIR> Downloads

...

0 File(s)

15 Dir(s)

C:\Users\foxr>cd Documents
C:\Users\foxr\Documents>copy * * \Users\zappaf
C:\Users\foxr\Documents>cd ..

Documents директорыясына өту
барлығын zappaf's директорыясына
көшіру
I деңгейге жоғары ауысу

C:\Users\foxr> cd Downloads	Downloads төменде ауысу
C:\Users\foxr\Downloads> move * * \Users\zappaf	Көшірудің орнына жылжу
C:.\Users\foxr> CD \Users\zappaf	Zappaf папкасына ауысу

ДОЖ эквивалентті болмайтынына назар аудару керек, сондықтан zappaf дейін жету үшін, сізге не салыстырмалы, не абсолютті жолды көрету керек\Users\foxr\ Downloads-тан zappaf-на дйін, сіз : CD ..\zappaf салыстырмалы жолын пайдалануыңызға болады.

C: Users\zappaf>dir Zappaf тізімдемесінің тізімі

Бұл енді Documents және foxr үшін Downloads табылған барлық файлдарды қамтиды.
Тізім төменде көрсетілген.

C: \Users \ zappaf> del * *
C.: \Users \ zappaf>

Өшірілген орынның неге ие болғанына назар аудару керек. Zappaf файлдардың өшірілуінен шектелген жоқсыз. Zappaf тізімдемесі үшін және файлдар үшін сәйкесіншк қорғалатын рұқсат құрылған жоқ.
Біз бұл мысалды осы пікірлермен аяқтаймыз. Linux сізде мында да, директорияны өшіріп және құру үшін сізде mkdir и rmdir. командалары бар. Mklink командасын пайдалан отырып, жұмсақ және қатта сілметке құруға болады.Өзіңіз аңғарғандай, * символы ДОЖ жұмыс істейді. ДОЖ командасы опцияларды жеткізеді. Del командасы пайдаланушыдан мүжбүрлі түрде del/p командасын пайдалануды сұрауы мүмкін және рұқсатсыз ақ /Q (жай тәртіп) командасы арқылы өшіріп тастауы мүмкін. Linux-тан қарағанда, del командасы рекурсивті тәртіпке ие болмайды. Файлдар, тізімдемелер және Файлдық жүйе 151

Файлдық жүйе және жүйелік әкімшінің міндеті

Жеке қолданушылардың өзіндік жеке файлдық жүйеде кеңістігі болсада, көптеген міндеттер жүйелік әкімші айналысады және басқарады. Бірінші және ең баста міндет ол файлдық жүйенің қол жетімді болуын біліп алу. Бұл жүйені қалауынша құрастыруы мүмкін. Linux mount және umount командалары файлдық жүйеде жаңа бөлімдер құрастыру үшін және оны бөлшектеу үшін қолданылады. Бұл команда сізден қандай файлдық жүйені және нүктені құрастыруды көрсетуіңізді та-

лап етеді. Құрастыру нүктесі бұл ат болып табылады, оның көмегімен қолданушылар соған сүйенеді. Мысалы, сіз `df-k` командасын орындай отырып, `/var` және `/home` қосқанда, әртүрлі құрастырылған нүктелерді көресіз. Арнайы `/etc/fstab` файлында тізіп көрсетілгендей, `mount-a` командасы файлдық жүйеде барлық компоненттерді орнатады (Файлдық жүйенің кестесі). Бұл команда жүйенің автоматты түрде инициализациясы кезінде орындалады, бір егер кейбір бөлімдерді бөлшектеу керек деп санасаңыз, онда сіз осы команданы (тамырлы сияқты) бөлшектеу үшін орындауыңыз мүмкін. `/etc/fstab` файлы өзіне қосымша ақпаратты кіргізеді, ол ақпарат пайдалы болуы мүмкін, мысалы, дисктік үстемелер, бұл файл кеңістіктегі тек қана оқылатын немесе оқу және жазба үшін арналған ба екен. Қазіргі кезде орнатылған бөлімдер `/etc/fstab` файлында көрсетілген.

Дисктік үстемелерді баптау өте қиын операция болып табылады, шектелген файлдық кеңістікте және көптеген қолданушылар бар жерде ол пайдалы болуы мүмкін. Сіз дисктік кеңістіктің басқарылуын қолданушылар үшін дисктік үстеме орнату арқылы, орнатып және оны жүзеге асыра аласыз.

Басқа амалдар өзіне файлдар шифрланған болуы керек немесе керек еместігін кіргізеді. Linux шифрлау үшін бастапқы ашық коды бар құрал ретіне ие, және де Windows үшін қол жетімді көпшілік домендерінің құрал реті болады.

Өшірілген сақтаулар осы компьютерге арналған жергілікті файлдық жүйесі бола алмайды, бірақ оның орнына желі бойынша қол жетімді. Өшірілген сақтаулар деректерін қолдау мақсатында, файлдық жүйені кеңейту қажет, сонда файлдық жүйе желілік бола алады. NFS осы таңда анағұрлым кеңейтілген түр болып саналады, және ол Unix/Linux жүйелерінде қол жетімді. NFS өте қиын болып саналады. Сіз NFS жайлы толығырақ кейбір оқуға арналған қосымша мәтіндерден таба аласыз.

Бұл соңғы тақырыптар пайдалы болса да, осы мәтінде қамтылмайды, бірақ Linux жүйелік әкімшілігінің мәтінінде қамтылады.

Компьютерлер тарихы ■ 201

1960 жылдың ортасында, IBM қызметкерлерімен өздерінің толымдаушы бірқатар бағдарламалық жасақтамасын өндіруді тоқтату туралы шешім қабылданды. Нәтижесінде, IBM компьютерлерін иемдеген ұйымдар, бағдарламалық жасақтаманы сатып алу үшін басқа бір іздеулеріне тура келді. Осылай, бағдарламалық жасақтама өңдейтін компаниялар енгізіле бастады, және де бағдарламалық жасақтама индустриясы пайда болды.

Бағдарламалық жасақтаманың тарихи аппараттық жасақтама тарихи-на қарағанда сондай қатты толқулы емес. Дегенмен, көптеген онжылдықтар қарсаңында өте маңызды жаңалықтар болды. Осы бөлімде олар қысқаша қарастырылады. АТ сүйене отырып, сіз бағдарламалар құра аласыз, бірақ бағдарламалық жасақтама үдерісі кезінде пайда болған мәселелер жайлы алаңдамауыңызға болады, өйткені сіздердің бағдарламаларыңыздың көпшілігі қысқа сықырлы бола алады. Дегенмен, бұл тарих біздің қазіргі кезде біз бұл бағдарламалауға қалай келдік, қайда орналасқанымызды көрсетеді.

Ертеректегі компьютерлік бағдарламалар машиналы кодпен, нақтырақ машина үшін жазылған, әдетте сол кезде, инженерлар жиі компьютерді өздері қосып, өздері құрған. Бағдарлама енгізулері жай ғана мәтіндік жиынтық емес, жад аудандарында қосылып болмаса, тізбекті есептеу кабель көмегімен жасалады, өйткені, ертеректе, қоңырауларды телефонды оператор байланыстырған. Бағдарлама орындалып болған соң, тізбекті қайта барлық компьютерде жалғап келесі бағдарламаға өту үшін қайта қосуға тура келетін. Еске түсіре кетейік, бірінші компьютерлер қызмет мерзімінің қысқарығына қарай сенімсіз болды және олардың вакуумдық құбырлары да сенімсіз болды. Осыған машина бағдарламалау кодының табиғатын және бағдарлама енгізудің әдістерін қоссақ (кабель арқылы), онда сіз өте қиын жағдайға тап боласыз. Тарихи анықтамаға жүгінсек, ең бірінші бағдарламашы өзінің бағдарламасын жұмыс істете алмады, бірақ оның вакуумдық құбырлары жұмыс істеп тұрды, логикалық функциялары дұрыс болды, бағдарлама дұрыс енгізілді. Тоқ аударғыш релеге қалай екенін белгісіз көбелек кіріп кеткен, сондықтан тоқ жібермей тұрған. Сондықтан, бұл «бағяғни, қате» деген терминға акеледі, қателікті білдіреді. Осыдан кейін біз, бағдарламадағы қателікті жою үшін дұрыстау термині ғана қолданбаймыз, жай ғана ақаулығын іздеуімізге болады. 7.14 суретін қараңыз.

9/9

0800 Antan started

1000 " stopped - antan ✓ { 1.2700 9.032 847 025
 1300 (032) MP - MC 1.98264000 9.037 846 995 correct
 (033) PRO 2 2.130476415 (2) 4.615925059(-2)
 correct 2.130476415
 correct 2.130676415

Relays 6-2 in 033 failed special speed test
 in Relay " 10,000 test.

Relays changed

1100 Started Cosine Tape (Sine check)

1525 Started Multi-Adder Test.

1545  Relay #70 Panel F
 (moth) in relay.

First actual case of bug being found.

1630/1650 Antan started.

1700 closed down.

Relay 3370

Сурет 7.14 Бірінші компьютерлік кате. (АҚШ-ң Әскери-теңіз тарихтық орталығымен берілген, сурет интернет кітапханасынан NH 96566-KH.)

202 ■ Ақпараттық технологиялар

Сонымен қатар, біз есептеуіш құрылғысының (I / O) енгізу / шығару командаларын еске түсіргеніміз жөн, негізінен перфокарт немесе магниттік таспалар және магниттік лентадағы жазу есептеумен шектелді. Кез-келген өндірілген шығарулар лентада сақталады, ол ленталық құрылғадан демонтаждалған, содан кейін принтерге орнатылған және басып шығарылған. Сөйтіп, компьютер бір кезекте бір ғана бағдарлама қоса алады, сондықтан, барлық енгізулер шығарулар шектеулі болған, және қандайда бір компьютердің операциялық жүйенің керек емес болған. Бағдарламашы сол бағдарлама өзінде енгізу-шығару кодтарын пайдаланған.

Бағдарламау кертартпасы бірнеше жаңартпадан кейін тіптен жай бола бастады. Бірінші идея ол бағдарламаны таратушы болды. Бұл бағдарлама деректерді енгізу ретінде басқа бағдарламаны алды және машиналы

кодтың нұсқасын шығарған, содан кейін компьютерде пайдалан бастауы мүмкін болған. «Бастапқы код» атымен белгілі түпнұсқалы бағдарлама, өзінің алғашқы түрінде орындала алмайды. Бірінші таратушылар «ассамблер» ретінде белгілі, ол ассамблерлі бағдарламаны машиналы кодқа аударған. Ассамблер тілі машиналы тілге қарағанда оңайырақ, оған өте нақты, түсінікті және төмен деңгейлі құрылым қажет (2 тараудың мысалын еске түсірейік). 1959 жылы аударма тілі әлдеқайда жақсарған, конвертерленген тілағылшын тілі ұқсатылып өте мықты бағдарламалық құрылыстармен жазылған. Жетілдірілген таратушы сыныбы «құрастырушы» ретінде және де тілдер «тілдердің жоғарғы деңгейі» атпен белгілі. Осылардың бірінші таратушысы тіл үшін құрылып FORTRAN деген атау ие болды. FORTRAN мағынасы, бағдарламашыға көбінесе математикалық формулаларға көрсеткен, олар алгебралық түсіндірілу командалық енгізулер , шығарулар және бақылау. Бақылау командасы (кезең, таңдаулар) ассамблер тіліне қатты ұқсаған, бірақ қалған тілдің бөлігі ағылшын және математика тілі сияқты деп есептелді. Тілдің атауы ТАРАтушы ФОРмулыдан тұрады. FORTRAN ең алдымен математикалық/ғылыми бағдарламауға арналған. Саудалық амалдарға арналған тіл шамамен сол кезде COBOL атымен құрылған (саудалық амалдарға арналған әмбебап тіл). Басқа тілдер жасанды интеллект салаларына (LISP), модельдеу саласына (Simula), және жолақпен (SNOBOL) жұмыспен жасауға арналып өңделген.

C code

```
for(i=1;i<=x;i++)
    sum=sum+i;
```

Assembly Language code

```
01231407  mov     dword ptr [i], 1
0123140E  jmp     main+69h (1231419h)
01231410  mov     eax, dword ptr [i]
01231413  add     eax, 1
01231416  mov     dword ptr [i], eax
01231419  mov     eax, dword ptr [i]
0123141C  cmp     eax, dword ptr [x]
0123141F  jg      main+7Ch (123142Ch)
01231421  mov     eax, dword ptr [sum]
01231424  add     eax, dword ptr [i]
01231427  mov     dword ptr [sum], eax
0123142A  jmp     main+60h (1231410h)
```

Machine Language code

```
C7 45 EC 01 00 00 00
EB 09
8B 45 EC
83 C0 01
89 45 EC
8B 45 EC
3B 45 F8
7F 0B
8B 45 E0
03 45 EC
89 45 E0
EB E4
```

Сурет 7.15 ассамблер және машиналы кодты салыстырудың жоғары деңгейі

Компьютерлердің тарихы ■ 203

Ассамблер және машиналы кодты салыстырудың жоғары деңгейі салыстырайық. 7.15 суреті жай ғана Си операторы бүтін 1-ден x мағынасына дейінгі соманың есептеуін көрсетеді. C коды бір операторды ғана көрсетеді: fog кезеңі. Кезең денесі өз алдына бір мәрте оператор болып саналады.

Кейде бір C командасы ассамблерлі болса ассамблер тілі (Intel x86 процессоры үшін) 12 командадан тұратын ұзын кодқа ие болады. Бұл суретте үш бағанды ақпарат түрінде көрсетілген. Бірінші бағанда команда да сақталатын жады мекен-жайы бар. Екінші бағанда, жұмыстың өзі ретінде, түртінді ұсынылған. Үшінші бағанда команда шығарып тастайтын операндтан тұрады.

Мысалы, бірінші команда да 1 саласындағы мағыналар бар, ол “dword ptr” ауысымға көрсетіп тұрады. Төртінші команда ең жеңіл “add eax”, түсіндірімнен тұрады, 1 мағынасы тіркелім деректеріне қосылып “eax” атымен беріледі. Әрбір ассамблер командасы бір машиналы кодтың командасына конвертеледі. Бұл жағдайда, машиналы тіл (суреттің оң жағында) командасы он алтылы жүйеге әкелінеді. Мысалы, бірінші ко-Бағдарламашыларға іздеу салынады!

Біздің қоғамда бағдарламалық жасақтамасының бағдарламашысы және инженері жаңа термин болады. 1962 жылы бірінші бағдарламалау университеттік факультет Перью Университетінде болды. 1965 жылы бірінші бағдарламалау саласындағы кандидат ғылымы Пенсильвания Университетінде алынды. 1970 жылдардан бастап бағдарламалау көптеген университеттерде енгізілді. Біздің тілде 1968 жылға дейін «инженерлі бағдарламалау» термині де болған жоқ.

Бүгінгі таңда, көптеген бағдарламашылар бағдарламалау салысын оқу дәрежесін алып жатыр, дегенмен, БЖ инженериясына арналған кейбір бағдарламалық оқу дәрежелері бар. Бірақ ақпараттандыру мамандығы пайда болғанша, бағдарламашылар кімдер болды? 1 тараудағыдай Джо тарихихындай, АТ маманы бағдарламашыға айналады, ертеректе бағдарламашылар өздігінен білім алған. Олар инженерлер немесе математиктер болған, немесе мекеме әкімшісі немесе бас есепші болған. Егер сіз қалай бағдарламалау керек екенін білсеңіз, онда сіз бірден бағдарламашы болар едіңіздер.

Қазір компьютерлік бағдарламашылар – өшіп бара жатқан түр. Шамалы компаниялар ғана бағдарламалау білітен біреуді жұмысқа ала алады, бірақ бағдарламалау негізінде білімі таяз. Демек, бүгін, сіз жұмыс бабында бағдарламалау тізімін көрсеңіз, сізден ақпараттандыру жөнінде оқу дәрежесі, ақпараттық жүйелер, компьютерлі инженерия немесе АТ керек болады.

манда «mob» 14 он алтылық мағындан жұтұрады, жұпты болып көрсетіледі. Ассамблер тілінің шифрлы бола алатынан түсіну үшін, ең алдымен жеңіл C кодын оқуымыз керек. Ассамблер тілінің түртіндісі әрбір операцияның не істеп жатқанын көрсетеді, бірақ машиналы код түсінік

бойынша толығымен көмескі. Оның орнына, С коды бізге «fog» сияқты ағылшын сөздері көмек беретіндігін айтады, ол өз алдына математикалық және жүйе символы ретінде жүреді.

Бірақ ассамблер тілі бағдарламашылар үшін шифрлі болмауы маңызды емес, бірақ бұл мәселе өңделуді талап етеді. Жоғары деңгейдегі тіл кодына қарамастан қай тілде аяқталуы үшін нүктелі үтір командалары ретінде, ағылшын сөздері, математикалық символдар, және таныс синтаксистан тұрады. С бағдарламалау тілі 1968 жылға дейін жазылмаған: бірақ сол мезгілдегі жоғары дейгейлі басқа тілдер (FORTRAN, COBOL және т.б.) кез-келген машиналы кодтың немесе ассамблерге қарағанда, түсінілуге өте оңай болып келеді. Жоғары деңгейдегі бағдарламалау тілі көшусіз болып келеді, бағдарламалық жасақтаманы әзірлеу мәселесі ғана емес, өйткені, дамыған шикі тіл бағдарламалық жасақтама өлшемін шектейді, бағдарламалау саласындағы ең маңызды жетістіктерінің бірі болып табылады. 204 ақпараттық технологиялар ■

Бұл машина тілін немесе құрастыру тілін жазуға мәжбүр етеді, 20 бағдарламашылардың командасы бағдарламалық жасақтама желілерінің миллиондаған түрін өндіре алуы екіталай.

1960 жылдың ортасына қарай, компьютерлер әлдеқайда қол жетімді бола бастады. Сонымен қатар, бұл көп кіріс және шығыс ресурстарын және компьютерлік желілер пайдаланушыларын мылқау терминалдар немесе шалғай жерлердегі компьютерге қосылуға мүмкіндік беретінімен пайдаланылады. Ал енді кезде, ол компьютерлер мен бағдарламалауды пайдаланып отырған жай ғана инженерлер емес. Ұйым ішіндегі бар компьютерге, кез-келген қызметкер компьютер пайдаланушы бола алған. Бұл қолданушылар, бәлкім, компьютер толымдаушыларын түсінбей, ешқалай компьютерді бағдарлай алмаған. Осы қосылған барлық қосымша бағдарламаға керек болды, және бұл қарапайым қолданушыға бағдарламаны іске қосу үшін жеңіл командаларды ензіруге рұқсат берген және файл деректерін ауыстыра отырып, қолданушылар толық бағдарламаны жазуға тиіс болмаған. Бұл бірінше операциялық жүйелерге әкеліп соққан.

1960 жылдардың басында операциялық жүйе «резидентті монитор» деп аталған. Ол әрқашан жадыда болады, кез-келген қолданушыны шақыру үшін қол жетімді болады. Ол кезінде «монитор» деп белгілі болды, өйткені ол қолданушылардың сауалын қадағалап отырды. Сауалдар бағдарламаны іске қосуға арналған мәнді түрде шектеулі болды, ол енгізген орынды (қайсысы ленталы файл немесе диск, қандай файлдар)

және шығару мағынасын (принтер, дисктағы файлдар, магнитті лентағы файлдар) көрсетті. Бірақ, 1960 жылдың ағымана сәйкес, қолданушылар компьютерлерге қол жеткізу мүмкіндігіне ие бола алған, резидентті монитор аса қиынырақ болуы тиісті. 1960 жылдың орта шегіне, резидентті монитор бағдарламаның – операциялық жүйесі деп белгілі боды, ол қолданушыға компьютерді басқаруды рұқсат еткен. Операциялық жүйе бағдарлама жоспарлауға (бірнеше қолданушылар бірнеше бағдарламалар сұралуы мүмкін), бағдарламаның орындалуына (орындалып жатқан кездегі бағдарламаның қосылуы мен мониторингі, жұмыс аяқталысымен бағдарламаны аяқтау) және қолданушылық интерфейске жауап берген. Операциялық жүйе бірнеше қолданушылардың сауалдарын бір уақытта өңдеуге мәжбүр. Бағдарлама жұмысы ең алдымен мультибағдарламалау негізінде орындалған, содан кейін уақытыл бөлінулермен (қазір көп амалдылық деп аталады).

Операциялық жүйе қолданушыны (басқа қолданушының ресурсын, екінші бір қолданушы бұзбауына келпiлдік берген) және желілік қауіпсіздікті қорғауға қызмет етеді . 1960 және 1970 жылдар аралығы бойы операциялық жүйе мәтінді негізде болды. Сөйтіп, қолданушы компьютердің толымдаушылар түсінбеуге тиісті болған және бағдарламалай жағдайында болған, ол операциялық жүйенің командаларын қалай пайдалану керек екендігін түсіну үшін, қолданушыдан назар талап етілген. VMS (виртуалды жады жүйесі) сияқты жүйелер DEC (Диджитал Эквипмент Корпорациясы) компьютерінде жұмыс істеген, VAX және JCL (тапсырмаларды басқаратын тіл) компьютерінде, IBM негіздік блогында жұмыс істеді, командалар жетілдіріліген және қиын болған, бағдарламалау тілі сияқты.

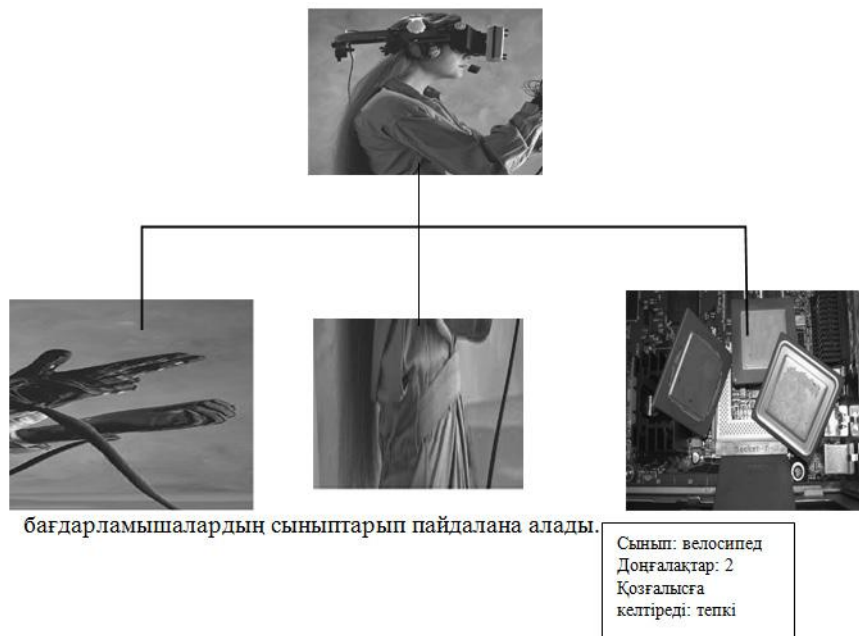
Дербес компьютердің дамуымен операциялық жүйе неғұрлым жеңіл болып қолданыла алды. Олардың арасындағы ең танымалы дисктік операциялық жүйе MS-DOS болды. Командалар диск жұмысын шектеу үшін мәнді дәрежеде болды – бағдарламаның қосылуы, файлдардың сақталуы, файлдардың жылжуы, файлдардың өшірілуі, тізімдемелердің құрылуы және тағы басқалары. Операциялық жүйе DOS деген атауы, қызық болғанымен, ол кез-келген дисктік немесе ленталы сақтау орын ретінде пайдаланылған. 1980 жылға дейін операциялық жүйелердің жаңа түрлері шықпаған. Дегенмен, сол уақыт аралығында

Компьютерлер тарихи ■ 205

1960 жылдары бағдарламашылардың алған сабақтарына сәйкес, 1970 жылы құрылымдық бағдарламалау сияқты танымал жаңа революцияға әкеліп соқтырды. GOTO сияқты өту командалары үлкен мөлшерде ертедегі тілдерде пайдаланылды. GOTO операторлары бағдарламашыларға

басқаруды кез-келген бағдарламалау жерінен, басқа да бір бағдарламаға жібере алды. Бұл бағдарламашылар үшін керемет нәрсе болды, әзірше бағдарламаның қателіктерін түсініп оны өзгерту үшін сіз бағдарламаны түсінбеуіңіз керек. GOTO командасының себебі «спагетти-код» деп аталатын кодыты құру түйінінде. Егер сіз бағдарламаны қадағаласаңыз, онда сізге нұсқаулықты бірізді тәртіппен қадағалау керек. Дегенмен, GOTO операторын қолданумен, сіз кез-келген командаға баса отырып, басқа бағдарлама орнына ауысып кетуіңіз мүмкін. Бағдарламаны қадағалау спагетти тәрелкесі сияқты көрінуі мүмкін. Құрылымдық бағдарламалауда, бағдарламашы жоғары деңгейдегі басқару құрылысымен шектеулі, «егер-басқаша» сияқты кезеңді шартты кезеңдер, және анайы GOTO операторын пайдалануға жіберілмейді. Бұл жаңа жоғары деңгейлі тілдер заманын ашты, C және Паскаль ең танымалы болып табылады.

1980 жылы бағдарламалау әлемінде тағы бір жаңалық шықты. 1980 жылдың ортасына дейін, бағдарламашы өмірде кейбір нәрселерді өңдеу, автокөлік сияқты физикалық нысаш болсын, немесе техникалық құжат сияқты абстрактілі нышан болсын, жеке ауыспалы нәрселерді қолданды. Ауыспалылар нышанның сипатын суреттеді. Мысалы, автокөлік үшін оның ауыспалылары ол: өзін қоса алғанда, бензин шығыны, автокөлік түрі, миль мөлшері және Көк кітап (сипатгізім журналы) бойынша ағымдағы бағалылығы. Ең жақсы үлгілеу амалы болмыс сыныбын анықтау үшін болған, олар нышан деп аталған. Нышандар бағдарламамен құрылуыға болар еді, әрбірі болашақ басқа физикалық нышанды үлгілеуде (мысалы, авокөлік сынбын ескере отырып, біз төрт әртүрлі автокөлік түрлендіре алуымыз мүмкін еді). Нышандар кейін бір-бірімен байланысты және басқа нышан түрлерімен сыбайлас болады. Нышанды-жөн сілтеу амалымен ескі амалдың арасындағы айырмашылық, негізінен ауыспалылығынан нышан өзі бір жеке болмыс болып келеді, ол өзінің жеке ішкі тәсілдерін өңдеу үшін бағдарламаландырылған болады, сонымен қоса, басқа нышаннан алынған хабарламалар. Көрсетілген сынптар арқылы бағдарламашы еншілес сынып анықтау жолы арқылы тілдердің кеңейте алады. Фмфұра деп аталатын технология арқылы бағдарламашы алдыңғы сынпты ала отырып осы сыныптан нақтырақ бір сынып құруына болады. Бұл екінші рет қолдану дәрежесін қамтамасыз етеді, кодты қайтадан керегінсіз жасау үшін бағдарламашылар басқа бағдарламышалардың сыныптары пайдалана алады.



Сурет 7.16 Нышанда-жөн сілтегіш мұрасы (тістегершікке негізделген/ фотосурет)

206 ■ Ақпараттық технологиялар

Мұра түсінігі 7.16 суретінде көрсетілген, мұнда бірнеше нақты велосипедтер үшін негізгі велосипед сыныбы болып табылады. Мысалы, иерархияда барлық велосипедтар тасу әдісін білдіреді, ол екі доңғалақты және адам тепкісінің нәтижесінде қозғалысқа ие болады. Дегенмен, велосипедтердің іштүрі болады, олар: жарыс велосипеді, әйел велосипеді және оқулы доңғалақтары бар велосипед.

Нышанды-жөн сілтеу бағдарламасы (НЖБ) ең алғаш 1970 жыл мен 1980 жылдың басында Smalltalk тіліне енгізілген. 1980 жылдардың ортасында Smalltalk нышанды-жөн сілтеу мүмкіндігіне жаңа C нұсқасы қосылды, ол C ++ деп аталды. C ++ соншалықты танымал болып кетті, басқа нышанды-жөн сілтеу бағдарламалық тілі (НЖБТ) 1990 жылы ғана енгізілді. Бүгінде әрбір бағдарламалау нышанды-жөн сілтеу бағдарламасының мүмкіндіктеріне ие.

НЖБТ –мен тұспа тұс қатар бірінші терезелі операциялық жүйе енгізілді. Нышанды-жөн сілтеу бағдарламасының қолданысын көрсету

үшін, Калифорниялық Ксерокс Пало Альто (Xerox Parc) жасанды ақыл зерттеушілері қабықшаны терезе негізінде құрастыды. Бұның ойы негізінен терезе нышан ретінде көрсетіледі дегенді білдіреді. Әрбір терезе кейбір жалпы белгілерге (өлшемі, орны, фон түсі) және операцияларға ие болады, олар кез-келген терезеде жұмыс жасай береді (оның ауысуы, оның өлшемінің ауысуы, оның бұзылуы). Нәтижесінде, өздерінің зерттеулерін жақсарту барысында, терезелі операциялық жүйе пайда болды. Олар өздерінің ойлап тапқан өнертабыстарын сатуды аса қатты ойлаған жоқ, бірақ келген кісілерге оны іс жүзінде көрсетуге дайын болды. Стивен Джобс және Стивен Возняк сияқты екі келушілер, осыны ерекше жаңалық деп тауып, бұл ары қарай бағдарламалауды өзгертеді деген болжам айтты. Джобс және Возняк өздерінің жеке ойларын іске асырамыз деп көп жылдар өткізді. Джобс бірінші Apple Lisa дербес компьютердің ГПО өңдеуінде қатысты. Бірақ 1983 жылы өнімнің бір бірлігі 10000 \$ болғанда. Олар өте аз сатылды. Шын мәнінде, Лиза жобасы шиеленіскен жағдай болды, сондықтан 1982 жылы Джобс осы жобадан кетуге мәжбүр болды, оның орнына ол Apple, Macintosh жобасына ауысты. 1987 жылы 2500\$ шығарылған Макинтош өте үлкен табысқа жетті.

*

Терезелі интерфейс көрсеткішті бағыттап отырып мәзір орнына немесе басудың және жылжытудың көмегімен, адамдарға компьютермен пайдалануға көмектеседі. Терезілі жүйемен VMS, DOS немесе Unix операциялық жүйе тілдерді оқудың керегі туындамайды, бірақ оның орнына қазір ешқандай оқусыз компьютерді ойша басқаруға болады. Сөйтіп, компьютерді барлығының қолдануына ашатын графикалық интерфейс болды. Қазіргі кезде, графикалық бағдарламалау және НЖБ бағдарламашы үшін ең мықты құрал болып табылады. Бүгінде таңда, барлық бағдарламалық жасақтамалар табиғаты жағынан графикалық болып келеді, және көптеген шығарылатын бағдарламалық жасақтамалар НЖБТ-нен тараған.

Бағдарламалауды пайдаланудағы тағы бір жаңалық ол құрастырушы емес, түсіндіруші болып табылады. Құрастырушы бағдарламалық жасақтама компоненттерін құрастыруға дейін анықталғанын талап етеді. Бағдарламалық жоба бірнеше ондықтан жүздікке дейінгі жеке компоненттерден құралғанда, бұл өте қиын болуы мүмкін, бұлардың бәрі құрастыруға дейін жазылуы керек. Мысалы, Java сияқты тілдерде ең алдымен басқа сыныпты шақыратын қандай да бір сынып жазылуы тиіс. Дегенмен, егер жай ғана ойыңызды тексергіңіз келсе, сіз оны толық кодсыз жасай алмайсыз.

* Аралаңыздағы орта жастағы адамдар бар болса, олар ең бірінші Apple

Macintosh жарнамалық роликті еске түсіруші еді. 1984 жылы қаңтарда Super Bowl XVIII желдету уақыты бірдей киінген, бірізді жолмен жүретін бір залда отыратын адамдардың жарнамасы \$ 1,5 млн көрсетті. Олар мүсінге ұқсас - «Үлкен Ағаны» естіп отырды. Ақ мәйке киген жас спортшы қыз дәлізге жүгіріп шығып экранға балғаны лақтыра салды. Сокқының салдарынан экран жарылды, және жарнама «... сіздер көресіздер, 1984 жыл ұқсас неге ұқсатынын » деген сөздермен аяқталып қалды. 1984 жылы жарнама режисер Ридли Скоттпен басқарылды. Сіздер бұл роликті YouTube желісінен таба аласыздар.

Лисп бағдарламау тілі 1950 жылдың аяғында жасанды ақыл зерттеулері үшін өңделді, оны түіндіруші пайдаланды. Ол бағдарламашыларға бір бағдаламаны бір дегеннен сынауға болатынын көрсетті, сөйтәп олар өздерінің бағдарламаларын бөлік бөлікпен құра алды.

Құрастырушы тілі мен түсіндіруші тілінің арасындағы негізге айырмашылық ол, түсіндірушілік тілң арнайы орта ішінде жұмыс жасайды, және түсіндірушілік деп аталады. Бағдарламашы команданы енгізеді, ал түсіндіруші сол команданы қайта құрып машиналы тілге айналдырып оны орындайды. Команда алдыңғы орнатылған, ауыспалы командаларға сүйене отырып , алдыңғы шыққан нәтижелерді қолдануы мүмкін, бірақ бұл міндетті емес. Сондықтан, түсіндірушілік бағдарламалу сессия түсінігене жүгінеді. Егер бағдарламашы тұспа тұс келген бірнеше команданы орындай алса, онда бағдарламаны оның бәрін бағдарламаға айналдыра алады. Құрастырушы тілде барлық бағдарлама орындалу қажет етілмес бұрын жазылуы мүмкін. Бағдарламаулау амалын табатын түсіндірушілікпен қанағаттанатын көп себептер бар: бірақ тиімді кодтың құрылуы олардің біреуі де жатпайды. Сондықтан көптеген бағдарламалау жобалары құрастырылады.

АҚ-әкімшісі ретінде сіздің бағдарламаңыз, сізден уақыт ара өзіңіздің жеке кодыңызды енгізуіңізді талап етері хақ. Бақытына қарай, тиімділік әркез мәселе бола бермейді (сіздің кодыңыз үлкен болмайды, бір бағдарламаға бірнеше команда), сіз түсіндірмелі ортаны пайдалана беруіңізге болады. Linux операциялық жүйесінде қабықша өзінің жеке түсіндірмелі ортасын ұстайды. Сондықтан, бағдарлама жазылуы, Linux сіздің файлдық командаларыңыздың ауысуының сұрағы ғана болмақ. Бұл шағын бағдарламалар сценарий (скрипт) деп аталады, ал бағдарламалау үрдісі қабықша командаларының сценарийі деп аталады. Windows операциялық жүйесінде, сіз DOS файлдарыңызды өңдеу пакеті үшін осындай нәтижелерді алу үшін жаза аласыз.

Скрипттарды өңдеу, бірақ та жүйелік әкімшіліктен тыс шығуы мүмкін.

Шағын скриптар әдетте желі басқарылуында, веб-сервер әкімшілігінде және дерекқор серверінің әкімшілігінде жазылады. Скриптар веб-әзірлеуші үшін құралдың таңдауы болып келеді, олар веб-сервердің (серверлік скрипттер) немесе веб-шолғыштың (клиенттік сценарийлер) қосылуы үшін шағын скриптар жаза бастайды. Серверлік скрипттер, мысалы, деректерді өңдеуде пайдаланылады, веб-бет түрінде енгізіледі немесе ақпаратты дерекқордан шығару үшін динамикалық веб-бет құрады. Клиенттік сценарийлер пайдаланушымен байланыс орнату үшін қолданылады, мысалы, форма дұрыс жазылғанын немесе компьютерлік ойын арқылы жазылғанын қамтамасыз ету жолы үшін. Бүгінгі күнде көптген түсіндірмелі тілдер осы мақсаттара үшін сценарий тілі ретінде жұмсалуды мүмкін. Сценарий тіліне Perl, PHP, Ruby, Python и asp кіреді. Біз бағдарламалауды және кейбір скрипт тілінің жазылуын толығырақ 14 тарауда қарастырамыз.

КОМПЬЮТЕР ПАЙДАЛАНУШЫ ЭВОЛЮЦИЯСЫ

Компьютер жабдығы және бағдарламалық жасақтама қалай әзірленсе, компьютерлі пайдаланушы да солай әзірленді. Ең ертедегі компьютер пайдаланушылары инженерлер болған, олар компьютерді құрастырған да, бағдарламалаған да. Ол пайдаланушылар электроника, электротехника және математика салаларында жоғарымамандандырылған білімдері болды. Осындай зертханалық машинада жұмыс жасайтын адамдар жүз шақты ғана болды. 1950 жылға қарай пайдаланушылар өршіп, өз қатарларына бағдарламашыларды қосты, олар ендігі кезде компьютер құрастыратын инженерлер болмаған (міндетті түрде).

1960 – 1970 жылы пайдаланушылар ұйымға қызметкерлерді қоса бастады, олар компьютерде уақыт өткізді және меңгерді. Бұл пайдаланушылар әдетінше өте білімді болды, бірақ инженер мен ақпараттандыру бойынша оқымыс болмаған шығар. Кейбір пайдаланушылар компьютерлік операторлар болды, олардың жұмыстары тасымалдаушы лентада ленталарды монтаждаудан және таза бөлмеде компьютер құрастыру жұмысынан тұрды.

Бірақ көптеген пайдаланушылар, оның орнына, мылқау терминалында өздерінің кеңселерінде жұмыс істеді. Бұл пайдаланушылар тобы үшін, не математика, не электроника және не бағдарламалау икемі кірмеді, бірақ көптеген жағдайда, олар өздерінің компьютерлік жүйесінде операциялық жүйе командаларын енгізумен оқытылған, өйткені, өздерінің амалдарын орындау үшін олар операциялық жүйе командаларын енгізе

білгендей. Төменде шамалы VMS-тан, өз алдына 1970-1980 жылдары танымал болған VAX DEC компьютерге арналған операциялық жүйеден мысалдар келтірілген (VMS кітабынан алынған үзінді)

```
ASSIGN DEV$DISK: [BOB] POSFILE.DAT FOR015
COPY SYS$EXAMPLES:TUT.FOR TUT.FOR
DEL SYS$EXAMPLES:TUT.FOR
PRINT/FORM = 1/QUEUE = SYS$LASER TUT.FOR
SHOW QUEUE/ALL SYS$LASER
```

1970 жылдары дербес компьютер әркімге компьютермен қолдануға мүмкіндік берді, олардың білімдеріне қарамастан. Ертеректегі компьютерлерді қадағалау үшін, пайдаланушыларға операциялық жүйеден бір нәрселер білу керек болды, олар әлбетте командаларды жазды. Бұл мәтінде біз жоғарыда MS-DOS командаларының кейбірін қарастырып кеттік. Бұл командаларды есте сақтау қиын болса да, олар меңгеруге оңайырақ, VMS сияқты операциялық жүйе үшін бұлар аса қиын.

1984 жылы шыққан Apple Macintosh операциялық жүйе командаларын қолдану ескірді. Командалық жолақта жұмбақты командаларды енгізудің орнына, пайдаланушы графикалық интерфейс көмегімен компьютерді басқарады. Барлық Macintosh бағдарламалық жасақтамалары ГБИ негізделді, ең алғаш, пайдаланушы қандайда бір арнайы мағыналарынсыз компьютерді пайдалана алды. Ол ойша оңай болды. Microsoft Windows операциялық жүйе бір жылдан кейін енгізілді, ал 1990 жылдарға қарай барлық компьютерлерде терезе негізінде графикалық интерфейс операциялық жүйе және жанама бағдарламалық жасақтама қол жетімді болды. Қызығушылық жағдайында қарастырсақ, жас компьютер үйренуді тез меңгере бастады, ал орта үлкен буын компьютерді үлкен, қымбат және қорқынышты деп ойлаған, ол әлі де компьютерді үйренуге екі ойлы болады.

Ең соңғы компьютерлік техника саласында өңделген құрылғылар өте сезімтал сенсорлы экранды болып келеді, олар мәселен, смартфондар және БҚ планшеттер болып табылады. Алғашқы енгізілген Apple компаниясының әртүрлі өнімі сенсорлы экран, компьютерді ГБИ терезелік әдісінен қарағанда, мықтырақ интуитивтік бақарумен қамтамасыз етеді. Айналдыру, тықылдатуғ пернетақта арқылы енгізу және саусақтарды пайдалану, өту, мүмкін операциялық жүйенің келесі қолданатын мүмкіндіктері болар. Microsoft жария еткендей, сыртқы түрдің бейімделуі және сезімталдылық келесі Windows 8 операциялық жүйесіне тән бо-

луы мүмкін.

Бүгінгі таңда, компьютер пайдалана білмейтін біреуді кездестіру қызық. Компьютерді пайдалану үшін қажетті білікті деңгей төмен деңгейде қала береді. Шын мәнінде, жер бетіндегі смартфондардың соншалықты мөлшерінен, шамамен халық санының жартысы компьютер пайдаланушылары деуге болады.

Компьютердің негізін түсіну үшін компьютер сауаттылығының кейбір негізін білу керек. Ал егер қиын түсінікті түсіну үшін, мәселен, бағдарламалық жасақтаманы орнату, толымдаушылауды монтаждау, өндірілімділік мониторингі және тағы басқалары, одан бетер көп білім қажет. Осының жартысын көптеген құрылыстар мен журналдар оқу арқылы оқып білуге болады. Бұны және де досың немесе отбасы мүшесінің біреуі жасап жатқанын қарап меңгеруге болады.

Бүгінде компьютердің қоғамға қатты әсер еткені, осының білімнің жартысы Интернет арқылы оңай қол жетімді, өйткені кейбір компаниялар мен жеке тұлғалар құрған веб-сайттар пайдалануға оқып меңгеруге көмектеседі. Тек тар шеңберлі білімдер мәселен, толымдаушыларды істеу, жүйелік/желілік әкімшілік, бағдарламалық жасақтаманы өңдеу және есептеу техникасы осылар терең білімді қажет етеді, мектеп қабырғасында немесе арнайы оқу барысында.

ҚОҒАМҒА ӘСЕРІ

Сіз көшенің төмен қозғалысымен кетіп бар жатқан автокөлікте отырсыз. Сіз тоқтау белгісіне келіп оның ауысып-жануын тосып тұрсыз. Сіздің GPS құрығыңыз сізге тура жүру керектігін көрсетіп тұр. Сіз жерсеріктік радио тындап келесіз. Тоқтау белгісі жанып тұрғанда, сіз досыңызға смартфон арқылы хабарлама жолдап жатырсыз (қазір кейбір штаттарда тыйым салынған). Сіз киноға бара жатқан жолдасыз. Сіз теледидардан ауа-райын қарап, жаңбыр жауатынын айтқанын естідіңіз. Сол кезде бүгін компьютерге мүлде жақындамасаңыз сіз толығымен компьютер технология саласына қаныққансыз.

- Сіздің автокөлігіңіз бірнеше процессорлардан тұрады: отынды бүркітін жүйесі бар карбюратор, тежеудің бұғатталуға қарсы жүйесі, аспап-құралдар панелі, бұның барлығы АЖ байланысатын бағдарламалық құрылғылар.
- Сіздің GPS бағдарламалық құрылғы, ол кіріс деректерін сенсорлы экраннан және орбиталық жерсеріктіктен алады.
- Сіздің жерсеріктік радиоңыз орбитадағы жерсерік арқылы жұмыс

істейді, ол зымыран және компьютермен қосылған, және елдің бір нақты обылысында дыбыс арқылы жеткізумен бағдарламаландырылған.

- Сіздің смартфон өз алдына кішірейтілген компьютер болып табылады.
- Сіз көрейін деп жатқан кино, бәлкім, мәтіндік процессорлар көмегі арқылы адаммен жазылған. Кино әлбетте сандық түрде дұрысталған. Арнайы эсерлер компьютерлік графика арқылы қосылған. Компьютер сияқты оның тізімі де кино өндірісінде пайдаланылған, киноның жарнамасын және маркетингін ескермей ол тізім ұзын болуы мүмкін.
- Ауа-райы арнайы жерсерік арқылы берілген (тағы да компьютер мен зымыранмен орбитаға шығарылған), ал ауа-райы үлгісі компьютерге қосылап жүктелген.
- Және тоқтау белгісінің шамы да компьютермен бақыланады. Жолда датчик көріп тұрсыз ба?

Біздің заманауи өмірде қандайда бір нәрсе байланысы компьютерсіз жүзеге аспайды деуге болады. Компьютерді өміріңізден біржола алып тастау үшін, сізге бір алыс шалғайдағы районға технологиядан бөлек жоқ жерге көшіп кетуге тура келеді. Дегенмен де, солай жасасаңыз сізге оның байланысы, есептер, пошталық бөлім, қала сыртындағы дүкенге бару, сонымен қоса, арқылы келе береді.

Компьютерді қолдану кез-келген және әрбір мансапта табуға болады. А-дан бастап Я-ға дейін, сіз мейлі есепші, жарнама, авиаұшулар, қарулы күштері, өнер, немесе зоология болсын компьютер бізге әрдайым көмектеседі, біз үшін көптеген жұмыс та жасап береді.

Бірақ байланыс біздің компьютерді жұмыс күші ретінде қолдансақ та шектелмейді. Олар экономикалық операциялардың алғашқы негізі болып келеді. Біздің барлық сатып алуларымыз электронды түрде жүзеге асады:

Мүлікті түгендеу, кредиттік карталармен байланыс, ақшаны есептеу. Компьютерлар біздің көңіл көтеру нысандарымызбен басым, бұл киноға құруға, теледидарлық бағдарламаларға, музыка мен өнерге жата ма, әлде жай ғана біз оларды тасымалдаушы арқылы көріп / естіп/ өнерді көребіз бе. Онымен қатарғ компьютер қазір біздің қарым-қатынасымыздың орта белінде жүр. Біз ұялы телефондарды телефондық қоңыраулар үшін пайдалансақ та, біз осы мобильды құрылғыны бір-бірімізді көру үшін, қадағалау үшін, аңду үшін, соңғы жаңалықтарды оқу үшін, ойын ойнау үшін де пайдаланамыз.

Сонымен Интернет бар. Интернет арқылы, біз сатып алу жасаймыз, инвестициялаймыз, оқимыз, білім аламыз, жаңалықтар көреміз, көңіл

көруді іздейміз, жанұямыз, достарымыз, таныс емес адамдармен байланыста боламыз. Жаһанды Интернеттің сипаты ол қол жетімділікте, заманауи қоғамда әрбір адамның өзінің дауысы болғанына бәрі жасалған. Кімді-кім өзінің пікірімен бөліскісі келсе, блогтар мен тақталар орнынан табылады. Әлеуметтік желілер бізге қашықтан достық қатынаста болуды және жаңа достар мен көңілдес табуға үлкен мүмкіндік берді. Интернет мәдениетаралық қатынаспен де қамтамасыз етеді. Енді біз басқа шет елде қалай тұру керек екенін көре аламыз. Біз олардың тарихы мен тарихи жерлерін көре аламыз. Біз сол елдерде YouTube арқылы болып жатқан жиналыстарды және көңіл көтерулерді бақылай аламыз. Және де, әлеуметтік желіліер арқылы сол елде қандай жаңалық болып жатқанын бірден біле аламыз. Мәселен, мысалы ретінде 2011 жылы арабтық көктем болды. Және сол қайта қандай жағдайлар орын алып жатқанын бар әлем бақылап отырды. Компьютердің әсерін көргіңіз келсе, келесі сұрақтарды қарастырыңыз:

- 1.Қашан сіз соңғы рет біреуге пошталық хат жаздыңыз? Қандай себеппен, құттықтау хаты немесе төлемнен басқа?
- 2.Қашан сіз соңғы рет досыңыздың үйіне өзіңіздің ұялы телефонынан қоңырау шалмай және электрондық хат жазбай қонаққа бардыңыз?
- 3.Қашан соңғы рет жаңалықтарды газеттен оқыдыңыз?

Сіз операциялық жүйе тарихын, DOS, Windows, Linux аспектілерін, және Интернеттің тарихын бірге қоса 8 және 12 тарауларда иегересіздер. Бағдарламалау тілі және оның кейбір эволюцияларын тағы қайта 14 тарауда қарастырамыз.

8. Операциялық жүйелердің тарихы

7 тарауда эволюциялық аппараттар құралдарын және бағдарламалық жасақтаманы қамтыдық. Ал бұл тарауда біз операциялық жүйенің эволюциясын Linux және Unix дамуының ерекше аспектілеріне, және жеке графикалық интерфейсті операциялық жүйенің негізінде ДК қарастырып, ерекше көңіл бөлеміз. Linux-ті игерудің екі жеке ағынын қамтымыз: біріншіден, бүгінгі күні бар, Linux дистрибутивтарының әртүрлілігін, екіншіден Linux дамуына ашық БЖ әзірлеушілерінің байланысын қарастырамыз.

Осы тарауды оқудың мақсаттары:

- операциялық жүйелердің қалай эволюция бірте-бірте дамығанын сипаттау.
- Linux-тің есептеу техникасы әлемінде қандай әсер қалдырғанын талқылау.
- Linux әртүрлі дистрибутивтарын салыстыру.
- ашық БЖ әзірлеушілерінің қоғамда қандай рөлі барын сипаттау.
- ДК үшін операциялық жүйенің дамуын қадағалау.

Бұл бөлімде біз ОЖ Windows және Linux негізіне сүйене отырып, әртүрлі операциялық жүйелердің тарихи мен дамуына қараймыз. Бұл екі операциялық жүйені атап айтамыз, олар әлемде кең таралған және танымал, және олар екі жеке бағдарлама жасақтамасының философиясын көрсетеді. Windows Microsoft компаниясының өнімі, операциялық жүйе болып табылады, коммерциялық тұғырнама ретінде әзірленген, дегендей компания меншігі болады, пайдаланушылардың қызығын арттыру үшін және сол арқыды көп қаражат табу шығарылған. Linux ОЖ Unix-тан тарап шыққан, Open Source қозғалысын қамтиды. Ол тегін, сонымен қоса, ОЖ ішіндегі ашық түрде қол жетімді, солай әзірлеушілер кодты арттырып және өзгертулеріне болады, және ОЖ үшін өздерінің жеке бағдарла-

малық жасақтамасын шығара алады, оған дейін олар Open GNUS (GL) лицензиясының келісімін қадағалайды. Ең алдымен қандайда бір ОЖ Windows және Linux қарастырмас бұрын, біз ертеректегі ОЖ эволюциясын қарастырамыз.

LINUX ЖӘНЕ WINDOWS АЛДЫНДАҒЫ

7 тарауда айтылғандай, ертеректегі компьютерлерде операциялық жүйелер болмаған. Бұл компьютерді пайдаланушылар компьютерді құрастыратын, бағдарламашылар мен инженерлер болған. Олардың бағдарламалары кішірек немесе мүлдем ресурссыз жұмыс істейтін болған (мүмкін, енгізу үшін перфокарта арқылы қол жеткізу, шығару дыбысымен магнитті лентада сақталатын). Перфокартадан оқылу операциялары және магниттік лентадағы жазбалар сол бағдарламаға қондырылуы тиісті болған шығар.

Кейбір өзгертулер ОЖ құруға қажетті қылуға әкеп соқтырды. Біріншісі тілдік аудармашыларды дамыту болды. Бағдарламаны іске қосу үшін, бағдарламашы алдымен өзінің бастапқы тілінен машиналы тілге бағдарламаны құрастыру керек болды. Бағдарламашы компьютерді бірнеше кадамға бағдарламанладыруына мәжбүр. Біріншіден, аударма бағдарламасы (түсіндіруші немес ассамблер) лентадан немесе перфокартадан енгізілуі тиісті. Екіншіден, бастапқы код тағы да лентадан немесе перфокартадан енгізілуі тиісті. Аударма бағдарламасы бастапқы кодтың жүктей отырып жұмыс істеуі керек, сосын бастапқы код аудары отырып, алынған орындаушы код магнитті лентада сақтады. Содан кейін, орындаушы бағдарлама магнитті лентадан енгізіледі, енгізілген деректерді тағы да картада немесе лентада сақтау. Бұл бағдарлама орындалып шығады және нәтижесі магнитті лентаға басып шығарылу үшін жіберіледі. ОЖ-сіз әрбір енгізілген және шығарылған амалдар бағдарламаның жартысы ретінде жазылу керек. Жеңілдету үшін, резидентті монитор осы операцияларды бірнеше команда негізінде ғана өңдеу үшін құрылған, бәлен мың жүздік пен ондық құрылымдармен емес. 8.1 суретіне қараңыз онда JCL тілінде суреттеледі (JCL IBM мэйнфреймасында қолданылады), файл бір жерден басқа көшіріледі.

ОЖ құруға қажетін туындатқан тағы бір өзгерту ол компьютерлердің болуы. Екінші буында компьютерлар қатты қымбат болмады. Нәтижесінде компьютердің көп сатылды және көп ұйымдар оған қол жеткізді. Компьютерлер мөлшерінің өсуімен, пайдаланушылар саны артты. Екінше буыннан басталған пайдаланушылар,а бағдарламашы немесе инженер болуыміндетті емес еді. Енді, қарапайым қызметкер компьютерді пайдалана алды.Операциялық жүйе өз алдына механизм болып келді, ол

адамдарға қажетсіз компьютерді пайдалануға және оны қалай бағдарламаландыру түсінігіне мүмкіндік берді.

Сонымен қатар, компьютерлік толымдаушылардың жақсаруы ОЖ екі тәсілге әкелді. Біріншіден, жұмыс қарқынының жылдамдығы, енді бағдарламалар жұмыс істеу үшін аз уақытты талап етті. Бірінші буында да компьютер пайдаланушы инженер және бағдарламашы болды. Бір бағдарламадан өту, бір пайдаланушыға өте көп уақытты алды.

```
//IS198CPY JOB (IS198T30500),'COPY JOB', CLASS = L, MSGCLASS = X
//COPY01 EXEC PGM = IEBGENER
//SYSPRINT DD SYSOUT = *
//SYSUT1 DD DSN = OLDFILE, DISP = SHR
//SYSUT2 DD DSN = NEWFILE,
//          DISP = (NEW, CATLG, DELETE),
//          SPACE = (CYL, (40,5), RLSE),
//          DCB = (LRECL = 115, BLKSIZE = 1150)
//SYSIN DD DUMMY
```

Сурет 8.1 JCL командалары

Орындалғанда аз уақыт кетіретін бағдарламаның көмегімен, осы өтуді жақсартатын ықылас ОЖ әкеп соқты. Сонымен қатар, компьютерлердің (вакуумды құбырлардың қысқа қызмет мерзімінен бас тарту) сенімділігінің артуы яқталуына дейін аса ұзын бағдарламалар орындауға мүмкіндік берді. Әлдеқайда ұзын бағдарламалармен бағдарламанландырудың қосымша ресурстарын өңдеуге қажеттілік туындады, оның арқасында ОЖ сұраныс артты.

ОЖ уақыттың бөлінуімен (көп амалдылық) және мультибағдарламанландыру дамуымен қиындап кетті. Бұл ондаған, жүздеген, мыңдаған пайдаланушыларға компьютерді бір уақытта қолдануға мүмкіндік берді. Бұл қолданушылықтың артуы операциялық жүйеден адан да бетер көп амалдарды талап етті, сондықтан ОЖ одан қиындап кетті. 7 тарауда жазылғандай, төртінші буындағы терезелі ОЖ пайдаланушылардың компьютермен қарым-қатынасын өзгертті. Төменде кейбір ертедегі операциялық жүйелер мен олардың үлесі жайлы жазылған.

Ең ертедегі «нағыз жұмыс үшін» қолданылған ОЖ енгізу/шығару GM-NAA болды, IBM 701 мейнфреймалы компьютерді қолдану үшін General Motors жазылды. Оның басты мақсаты алдыңғы бағдарлама жұмысын аяқтаған соң, автоматты түрде жаңа бағдарламаны орындау болды. Ол қайта өңдеу пакетін қолданды, онда енгізу бағдарламен берілді. Ол 1995 жылы жазылған резидентті монитордың кеңітілген жинағы болды,

ол бағдарламалармен қатар қосылған мейнфрейманы енгізу және шығару құрылғыларына қол жеткізе алуы мүмкін болды.

Манчестерлік Университет үшін Atlas Supervisor 1957 жылы бір уақытылы пайдаланушылар қолжетімділігін кеңейтті. Ол бірінші нағыз ОЖ болды (резиденттік монитор емес). Параллельді өңдеу 1960 жылдың басына дейін қол жетімді болмады (CTSS қар.). Сонымен қоса, вертуалды жадыны ұсынған алғашқы Atlas болатын.

BESYS Bell ОЖ болды, 1957 жылы мейнфреймов IBM 704 үшін Bell Laboratories-пен әзірленген. Ол кіріс дыбысын перфокарттан жіне магнитті лентаға дейін әзірлеген, сонымен қатар, кез-келген ақпаратты принтерге немесе магнитті лентаға енгізеді. Ол FROTRAN —ға құрастыру үшін және бағдарламаны іске қосу жасалды.

IBSYS 1960 жылдан бастап, IBM компаниясының IBM мейнфреймаларымен 7090 и 7094 шығарылды. ОЖ командалары бағдарламаларға орнықты, ол әрбір ОЖ командасының алдына \$ белгісін қойды, ол оны FROTRAN командасынан айырып тұрды. Осындай ыңғай нәтижесінде JCL құрамын мейнфреймді IBM 360 и IBM 370 бағдарламасын іске асыруға мүмкіндік берді (сурет 8.1)

1961 жылы CTSS немесе уақытты бөлетін үйлесімді жүйе Массачусеттегі есептеу технологиясы орталық институтында (MIT) шығарылды, ол жүйе уақытты бөлетін бірінші шын ОЖ болды (көп амалды). Atlas Supervisor қарағанда, CTSS компоненттерінің біреуі пайдаланушылық үрдіске жауап берген, әр біріне үрдістік уақыт үлесін ұсынған (осында уақыт бөлімі деген атау келген).

EXEC 8 1964 жылы мейнфреймов Remington Rand's UniVac үшін дайындалды. Ол бірінші табысты коммерциялық көп процессорлы ОЖ болды (бірнеше процессорда жұмыс істейтін операциялық жүйе). Ол бірнеше әдісті басқаруды үрдісті қолдады: бөлшектенетін, бөлінетін, уақыттың бөлінуі және нағызғы уақытта өңдеу. Соңғысы көрсетіп тұрғандай, үрдіс баяу жұмыс істейді, кідірусіз және берілген мерзім аясында аяқталады, пайдаланушымен нағызғы уақытта байланысады.

TOPS-10 1964 жылы Digital Equipment Corporation компаниясымен (DEC) мейнфреймді PDP-10 сериялары үшін шығарылған. TOPS-10 уақытты бөлінудің басқа танымал ОЖ болып табылады, өйткені ол ортақ жадыға беріледі. Ортақ жады бірнеше бағдарламаларға бір-бірімен жады арқылы қатынауына мүмкіндік береді. Қолданылатын ортақ жадыны көрсету үшін көп пайдаланушылық Star Trek негізіндегі DECWAR атты компьютерлік ойын әзірленді.

1964 жылы MULTICS бағдарламалық кодтың динамикалық байланы-

сын енгізді. Бағдарламалық инженерияда ішкі бағдарлама кітапхананы шақыру үшін, ол бағдарламашылар үшін ортақ болып табылады (басқа бағдарламашылармен жазылған, құрастырылған және кітапханада сақталған кодтың бөліктері). Динамикалық байланысты кодтар арасында қолданғанда, бұл үзінділер жадыға қажеттілік туындағанда ғана жүктеледі. MULTICS осыны бірінші болып ұсынды. Бүгінде біз Windows ОЖ осындай тәсілді «DLL» файлдың қолданылғанын көреміз. MULTICS модульдер негізінде әзірленген ОЖ болды, ол көптеген әртүрлі аппараттық тұғырнаманы қолдай алды және көлемді болады, жүйеге қосымша ресурстар енгізілген кезде, ол әлі де тиімді болды. Сонымен қоса, MULTICS жаңа тұжырымдамалар көрсетті, оған қол жетімділік тізімінің бақылауы, файлдардың қол жетімді бақылауы, үрдістер жадысын және файлдық жүйені бөлді, жіне иерархиялық файлдық жүйені кіргізді. MULTICS коммерциялық табысты ОЖ болмады, бірақ қолдануда бірнеше ондаған жылдар бойы сақталды, сонда 1980 жылға дейін кейбір ұйымдармен қолданылды. Бұл ең ықпалды ОЖ бірі болды, өйткені ол келесі Unix ОЖ үшін оның негізі болып қаланды.

OS360 IBM 360 мейнфреймы (кештеу нұсқасы үшін OS370 для IBM 370) үшін 1966 жылы шығарылды және ОЖ бастапқы партиясы болды, содан кейін мультибағдарламаландыру қосылды. Дегенмен, қандайда бір амал көп амалдық көмегімен орындалуына мүмкін (іс жүзінде көп ағынды болды, үрдіс көп амалды болды, бірақ үрдістер арасында ешқандай көп амалдылық болмады). OS360 JCL енгізу/ шығару командалары үшін қолданды. OS360 басқа операциялық жүйелермен енгізілген жаңашылдықтар ретін бөлді, ол виртуалды жады және иерархиялы файлдық жүйе, бірақ өз алдына жеке виртуалды әдіс қорына қолжетімділігіне ие болды (кейін ол дерекқор сақтау үшін негіз болды), сонымен қатар, ол еншілес үрдіс қабілеттілігін арттырды. Келесі OS360 деректерді жіберу нышаны болды, ол ОЖ кез-келген түріндегі терминалымен қатынас орната алуына мүмкіндік берді. IBM 360 және 370 мейнфреймендерінің танымалдылығынан, OS360 өз заманындағы операциялық жүйелердің ішіндегі ең танымал болды. IBM 370 үшін түрлендірілген OS360 өзі де бірнеше түрлендіруге ие болды. Бүгінде OS360 ең танымал тәжірибелі ОЖ болып қала береді және де еркін жүктеу үшін қол жетімді.

UNICS (кейінірек, UNIX) AT&T 1969 жылы (келесі бөлімді қараңыз) әзірленді. Сонымен қатар, 1969 жылы Авиалинияны басқаратын бағдарлама (ACP) IBM автоматизация жүйесінен IBM авиакомпаниясына арналған бөлек жеке бөлім болды, ол авиабилеттерді брондауды өңдеді. Бөлімнен кейін, ACP, кейінірек ретінде TPF белгілі болды, (тарнсакция өңдеу құрылғысы) ОЖ транзакциялар өңдеу транзакциялану үшін де-

ректар базасын авиакомпанияның өндеу тапсырмалары, кредиттік карта, сондай-ақ қонақ үйлер мен автокөлікті жалдау мәселелері болды. Бірақ бұл инновациялықтың өзі болды, бұл мысал ОЖ үшін арнайы міндеттер ретінде сондай-аппараттық тұғырнама қажет болды.

1970 жылдың басында көптеген операциялық жүйелердің жай мүмкіндіктері кеңейтілді, енгізілген күні 1960-шы жылдардың ішінде болды. Көрнекті ОЖ, VM үшін IBM пайдаланылады және 1972 жылы шығарылған виртуалды машиналарды құруға мүмкіндік береді.

UNIX тарихы

Unix ОЖ қалануы 1960-шы жылдың соңында болды. Бұл ең қуатты және портативті операциялық жүйесі болатын. Оның мықтылығы әртүрлі ерекшеліктерден тұрды: әкімшілендіру файлдық жүйесі, күшті желілік компоненттері, қауіпсіздік, бағдарламалық жасақтаманы тапсырыс бойынша орнату, сондай-ақ өзінің жеке бағдарламасын анықтау мүмкіндігі, қабықша және қоршаған ортаға бейімделу скриптары. Unix жартылай мықтылығы ОЖ кіріс дыбыстары үшін командалық жолақтың ұсынған иілгіштігінен болады.

Командалық жолақ пайдаланушыға немесе жүйелік әкімшіге командаларды енгізу үшін үлкен нұскалардың түрлісін ұсынады. Бұл портативті, өйткені ол C тілінде жазылған, ол барлық тұғырнамаларды құрастыра алады, және суперкомпьютерді, мейнфреймді, жұмысты бекеттерді, дербес компьютерлерді және ноутбуктарды. Шын мәнінде, Unix соншалықты танымал және табысты болды, ОЖ Macintosh үшін де (бірақ Mac терезелі жүйесі бірінші орын алады).

Unix AT & T Bell зертханасында құрылды. Ерекше ОЖ қолдану PDP-7 қамтылды, қызметкерлер осы машинада ойын ойнау үшін (Space Travel атты). Unix ассамблерде жазылды және портативті болмады. Ең алғашында ол табысты емес ОЖ болды; шын мәнінде, онша көп жүйе болмады. Бастапқы енгізуден кейін, Bell зертханасының екі қызметкері Кен Томпсон және Деннис Ритчи Unix-ті нығайтуды бастады, ол файлдармен жұмыс істейтін тәсілдерді қосты (көшіру, өшіру, дұрыстау, басып шығару), және пайдаланушы командаларды бірінен сон бірін енгізу үшін интерпреторлы командалық жолақ қосты («жұмыс» атты перфокартты серия арқылы емес). 1970 жылы ОЖ ресми түрде Unix деп аталды.

1970-жылдардың басында Unix іске қосу үшін арналған PDP-11 қайта өңдеді, PDP-7 машинасынан қарағанда әлдеқайда танымал. Оның қай-

та жазылып шығуының алдында, Ритчи бірінші болып C бағдарлама-лау тілін өңдеп және енгізді. Ол ОЖ іске асыра алатын тілді әдейі әзір-леп шығарды, содан кейін PDP-11 үшін Unix C көмегімен қайта жазып шықты. Бұны жазып шығуда басқа қызметкерлер жұмылдырылды, олар Unix жаңа функциялар енгізді олар құбыр деп аталды (олар 9 тарауда қарастылады). Операциялық жүйе кейіннен басқа фирмалар арасында тарап кетті (тегін емес, төлену үшін).

1976 жылы Unix бірнеше компьютерге жазылып шықты, және Unix тобында тіркеліп, ОЖ қолдау мен талқылауға ие болды. 1976 жылы Bell зертханасынан академиялық демалыс алды, Беркли қаласындағы Калифорния университетінде сабақ берді, ол UCB студенттерімен Unix Беркли нұсқасын әзірледі, қазір ол BSD Unix деп белгілі. BSD 4.2 өте күшті танымал релиз болды. Осы мезгілдің жетегінде, Unix TPC/IP хаттамасын қабылдайды, сөйтіп, компбьютерлер желі арқылы дерек-терімен алмаса алды. TPC/IP ARPANET(Интернет болды) қолданған хаттама болды, осыны қамтамасыз ету үшін, Unix-та мында болды.

1980-ші жылдары, өсіп келе жатқан Unix танымалдылығы әлсізденді және оған операциялық жүйе таңдау үшін көптеген компаниялар сатып алатын мейнфрейм және мини-ЭЕМ қажет болатын. Қымбаттылығы мен қиындығына карамастан, ол ең жақсы қол жетімді таңдау болды. 1988 жылы ашық бағдарламалық жасақтама құрылды (OSF) және ОЖ бағдарламасы еркін бағдарламалық жасақтама ретінде ашық ниетті бол-ды. «Тегін» термині сіз ойлағандай емес. Қалаушы Ричард Столлман «тегін» дегенді бостандықтағы «сыра» емес дер еді. Unix пайдаланушы-лары қоғам өздерінің уақыттарын жаңа операциялық жүйе әзірлеумен айналысқан және бағдарламалық жасақтаманы қолдау үшін кез-келген оны алуға, қолдануға, өзгертуге болған. Нәтижесінде әзірленіп басылып шығарылған нәрсе тек «тегін» тұжырымдама аясында шығарылған. Ол барлық бастапқы код тегін түрде қол жетімді болғанын талап етті.

* GNU Unix емес GNU деп шифрланады. Бұл рекурсивті анықтау, бұл шын мәнінде олай емес. Бұны Столлман беруге тырысты, яғни опера-циялық жүйе емес, Unix-ұқсас операциялық жүйеден тұратын тек еркін бағдарламалық жасақтама болды.

Ол GPL (GNUS) әзірлеп шықты, ол айтты, кез-келген бағдарламалық жасақтама GPL бағдарламалық жасақтан әзірленеді деп, және де GPL лицензиясы шегінде басылып шығарылу керек, сөйтіп, еркін қол жетім-де болу үшін. Бұл Unix қауымында үлкен өзгерістер енгізді, біреу тегін жұмыс істеуге дайын болды, біреулер табыс табу мақсатында жұмыс

істеді.

OSF әлі күнге дейін бар жіне әлі күнге дейін тегін бағдарламалық жасақтама шығарады, ал кейбір компаниялар өздерінің жеке нұсқаларын Unix істеп сатады. Unix бақылайтын AT & T Novell берілді, содан кейін кейінірек SantaCruzOperation берілді, ол кезде басқа компания-өндірушілер Unix Sun кіргізді (ол Solaris атты нұсқа шығарады), Hewlett Packard (HP-UX сатады), IBM (AIX сатады), и Compaq (Tru64 Unix сатады). Unix бүгінде танымал, бірақ бәлкім Linux сияқты емес шығар.

LINUX ТАРИХЫ

Linux Unix өте қатты ұқсас, бірақ Minix атты ОЖ негізінде «ойыншықты» еркін түрде қаланды. Minixm Кітап авторы Эндрю Таненбаума ОЖ үлгісі ретінде қол жетімді болды. Хельсенки Университеті студент Линус Торвальдс ОЖ терең зерттегі келді, бірақ ол өзі Minix оқысада, ақыр соңында оны орташа деп есептеп, өзінің жеке нұсқасын жазуға кірісті. 1991 жылы 28 тамызда ол comp.os.minix Usenet жаңалықтар тобына жазды, адамдар оның ниетін көріп, нәтижесінде қызығушылық танытар деген ойда болды. Оның алғашқы өтінішінде ол Intel 386/486 (IBM AT) процессоры үшін өзінің Minix нұсқасын әзірлеп жатқанын көрсетті. Ол бұнымен хобби ретінде айналысқысы келді, және оның алғашқы хаты басқа адамдардан жинау талпысы ретінде болды, олар осы ойынға қызығушылық танытқандай. Оның алғашқы іске асырылуы Bash түрлендірушісі және C құрастырушы , және ағымдағы GNU, GCC жобасы. Bash Bourne Again қабықшасы болды, ол Bourne қабықшасы, Unix үшін қол жетімді болып жасалған. Bash өте танымал қабықша болып табылады ол жайлы 9 тарауда білетін боласыздар. Ол тағы бастапқы іске асыру толығымен кез-келген Minix кодтан еркін болды. Дегенмен, ертедегі ОЖ кез-келген дискті іске қосатын драйверлер құрылғысы жетпеді, қатты диск AT басқа.

Торвальдс 1991 жылы қыркүйекте 01 нұсқасын және 1991 жылы қазанда 02 нұсқасын шығарды, және пайдаланушылардан көптеген ескертулер алды, өйткені олар жүктеп сынағанда ОЖ ысылмаған болып шықты. 5 қазанда тағы жаңа нұсқаны ұсыну үшін Торвальдс сол жаңалықтар топбында тағы хабар қалдырды. Мұнда ол операциялық жүйе көптеген әртүрлі Unix бағдарламалары іске қосалатынын жеткізді, оған Bash, GCC, СЭД (10 тарауда айтылады), сығымдау да кіреді. Ол бастапқы кодты қол жетімді етті, басқалары тек қана жүутеп және ойнау үшін емес, кодпен жұмыс істеу үшін, ол кезде өте үлкен қабілеттіліктерге ие

бола алады.

Желтоқсанға қарай, Торвальдс тағы да 10 нұсқаны шығарды, ОЖ әлі шикі болғанына қарамастан. Мысалы, оның Bash қабықшасына енгізу арқылы жүктелетін қабілеті болмады, және ол тек қана IBM AT қатты дискісін ғана қолдап тұрды. Ал 11 нұсқасы бірнеше құрылғыларды қамтыды, ал 12 нұсқада Торвальдс ойлады, егер операциялық жүйе айтарлық пайдалы және тұрақты болуы керектігін түсінді, және ол 95 нұсқаны ақтап шығу керек. Қазіргі кезде Торвальдс Танненбаумдан жауап естіді, тұжырымдама ретінде, ол «Linux ескірген» деді. Таннебаума өзінің пікірлерінде қателескен сияқты. Өзінің жеке бірінші толық көлемді нұсқасын шығара отырып, Линус Торвальдс Жалпы Азаматтық лицензия GNU еремін шешті. Бұл кез-келгенін ОЖ тегін көшірмесін алуға мүмкіндік берді, онда өзгертулер енгізуге және осы өзгертулерді тегін жариялауға мүмкіндік берді. Сонымен қоса, Торвальдс басқаларына Linux нұсқасын шығаруға рұқсат берді.

Бұл шешім тегін нұсқасы және коммерциялық нұсқасы кездосок ой болып шыққан екен.

Бірнеше жыл ағымында Linux жақтастарының тобы жүз мыңмен есептелді. Бағдарламалық жасақтаманы пайдалы және қалаулы етіп жасау үшін, пайдаланушылар қауымдастығы өнің үлесін қосуға тырыса берді, ол кезде коммерциялық нұсқа әзірлене берді. Ортақ Linux бағдарламауын айтарлықтай пайдалы ету үшін, графикалы пайдаланушылық интерфейстер қосыла бастады.

Бүгінгі таңда, Unix та Linux та ОЖ айтарлықтай танымал түрлері болып есептелді. Екеуі де көлемді спектр машиналары үшін, қалталы құрылғыдан бастап ноутбукка дейін және дербес компьютерден мейнфреймді суперкомпьютерлер үшін қол жетімді болды. Шын мәнінде, бірақ Linux келесі барлық машиналар/процессорлар үшін жұмыс істейді: Sparc Sun компаниясы, Alpha Compaq компаниясы, процессор MIPS, ARM процессоры (басқа да портативті құрылғыларда), x86 Intel / Pentium дейін, процессор PowerPC, Motorola компаниясы, 680x0 процессор, RS6000 IBM компаниясы, және басқалары. ОЖ соншалықты танымал екенінің себептері, бұл көптеген серверлердің Интернетте жұмыс істейтін Unix немесе Linux нұсқалары.

Міне, бірнеше Linux туралы қызықты фактілер:

- әлемдегі ең қуатты суперкомпьютер Sequoia IBM, және 446 500 үздік супер компьютерлер Linux нұсқасында жұмыс істейді.
- Тоқсан бес пайыз серверлер пайдаланылатын голливудтық анимаци-

ялық студиялар Linux-та жұмыс істейді.

- Google өзінің веб-серверлері Linux-та іске қосады.
- USS Android үшін Google-NokiaMaemo салынып, үстінен Linux өзегі құрылды.
- Айта кету керек, 33,8% барлық серверлері Linux-та жұмыс істейді (2009 ж.), ал тек 7,3% - ға Microsoft серверінде жұмыс істейді.
- Қол жетімді бағдарламалық жасақтаманы Linux қамтиды, MozillaFirefox және OpenOffice өзіне қосады, олар тегін болып табылады және AcrobatReader және AdobeFlashPlayer жеке меншік болып табылады.
- 300-ден астам бөлінулер Linux жан-жақты қамтылған, олар бүгін қолданылады.
- Debian Линукса нұсқасы 280 миллион астам код жолақтарынан тұрады, егер олар коммерциялық негізде \$ 7 миллиард артық болады; өзіндік Linux ядросы (1.0.0) 176,250 кодты жолақтардан тұрды.

Басқа жағынан алып қарасак, 2010 жылдың қаңтарына қарай барлық жұмыс үстелдерінің 1% Linux жұмыс істеді. Linux танымалдылығына қарамастан, оның кросс-платформалы мүмкіндіктері мен Linux нұсқасын алу қабілеті, және қолданбалы бағдарламалық жасақтама тегін болы, сонымен қатар, бірнеше басқа Linux тартымды сипаттамалары бар. Біріншіден, Linux тұрақты ОЖ болып саналады. Егер сіз Windows пайдаланушысы болсаңыз, бағдарламалық жасақтаманың еш себепсіз және ескергісіз құлағанын көріп, қапаланар едіңіз. Тағы да басқа жағдайлар туындауы мүмкін, өйткені операциялық жүйе тым ұзақ жұмыс атқарғандықтан, Windows жүйесін қайта жүктеу керек (Windows қайта жүктеу қажеттілік себептері операциялық жүйе беттерінің және фрагменттелген пейджинг кеңістігінің бүлінгенін болады). Linux әдетте ешқашан қайта жүктеуді талап етпейді. Шын мәнінде, Linux тек тоқтап тұрады, ақау болып жатқан жағдайда пайдаланушы жай ғана жүйені жауып тастайды. Жаңартуды талап етпеген жүйе, қайта жүктеуді керек етпейді. Сонымен қатар, Linux компьютерлік вирустармен жұқтырылмайды.

Көптеген жағдайда тек Windows ОЖ вирусты жұқтырады, ал кімде-кім Linux жүйесін бұзғысы келсе, оны сәтсіздік күтіп тұрады, өйткені Linux жадының көмегімен басқарылатын бағдарлама, қандай да бір бұзылудан қорғайды және оған кепілдік береді (4 тарауды қараңыз). Сонымен қатар, Linux юағдарламалық жасақтамаға өте тиімді қолдау көрсете алады. Эксперимент ретінде, Oracle Oracle 9i өнімділігін Windows 2000 ңске қосқандағы Linux салыстырды. Linux орындаулары 25-30% тиімдірек. Келушілерден сұрау бойынша, 2011 LinuxCon ең тиімдісі Linux Ubuntu болды, (34%), содан кейін Fedora / Red Hat (28%). Сұралғандардың тең

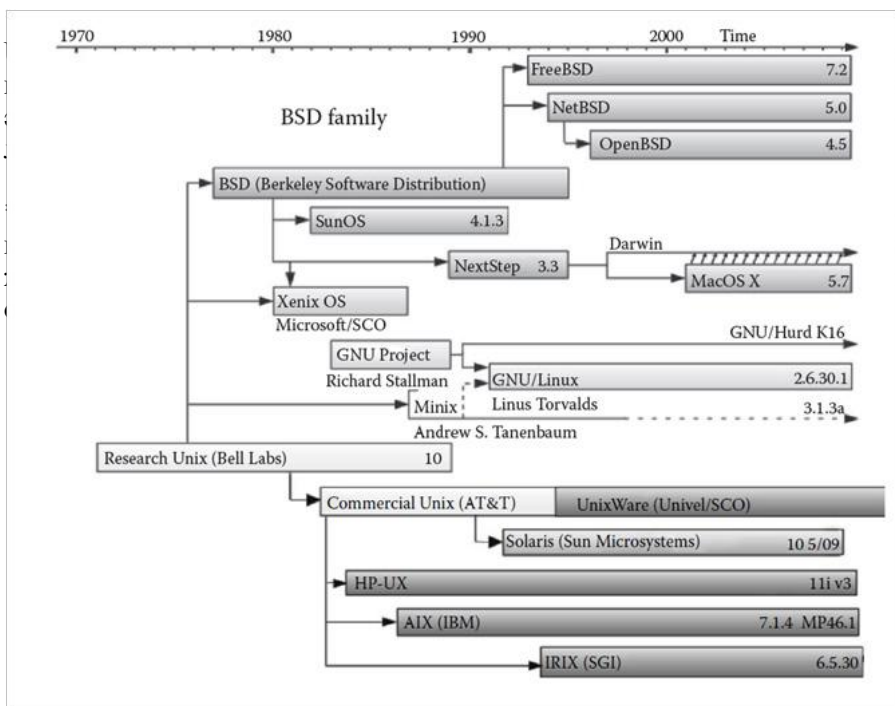
жартысы Linux үйде, жұмыста, мектепте, қолданады, ал 38 % пайдаланушылар Linux жұмыста ал 14 % үйде қолданады екен.

ЕРЕКШЕЛІКТЕРІ МЕН ТАРАЛУЫ

8.2-суретте әртүрлі релиз кестесі Unix, MacOS Linux-релиздері (2010 жылға дейін) арқылы түпнұсқалы Unics ОЖ арасындағысы көрсетілген. Маңызды релизді дистрибутивтарды көру үшін 8.3 суретін қараңыз. Unix және Linux арасында қаншалықты айырмашылық бар? Linux түрлендірулерінің арасында қаншалықты айырмашылықтар болады? Неге соншалықты көп релиздер бар? Неге сол соншалықты абыржытады?

Unix және Linux арасында айырмашылықты қарастырайық. Егер сіз екі операциялық жүйеге қабықша ретінде қарасаңыз, екеуі өте ұқсас екенін байқайсыз. Ол дерлік бірдей командалардан тұрады (мысалы, Ls, CD, RM, PWD) және де осы командалардың көбісі қиыспалы және бірдей параметрлерге ие. Олардық екеуінде де қол жетімді қабықша және кең таралған графикалық пайдаланушылық интерфейстері бар (GUI) олар екеіді тұғырнамада (KDE және Gnome) жұмыс істейді. Олар өте ұқсас жоғары деңгейлі тізімдеме құрылымына ие бола алады. Сонымен қоса, Unix және Linux сенімді, басқа әртүрлі аппараттық амалдар үшін портитивті, және кішкентай көлемді (ОЖ Windows түрінен қарағанда). Онда айыршалығы қандай?

Linux ашық бастапқы кодты болса, басқа бағдарламалық өнімдер, Unix және Linux бастапқы кодымен жазылған, Unix коммерциялық нұсқасы ашық болмайды, және Unix нұсқаларының көпшілігі коммерциялық өнім болып табылады. Unix шығарылымдары үшін стандарттар бар, олар нұсқалар арасындағы байланысты қолдап жалғап тұрады. Бұл Unix бір нұсқасы үшін жазылған өнім басқа нұсқада жұмыс істеуіне кепілдік бермейді, бірақ бұл шын мәнінде көмектеседі. Linux үшін, Linux Standards Base жобасы 2001 жылы басталды және Linux типті релиздер үшін әрқашан көмектесіп тұратын стандарт болып саналады. Стандарт тарамдар құрауға тиісті ма тиіс емес пе соын суреттейді. Мысалы, стандарт айтады, құрастырушының қандай нұсқасы C++ GNUS және қандай библиотекалар C ++ қосылуға тиіс екенін. 2011 жылы Java құраушысы стандарттан шеттеліп тастады. Дегенмен, Linux бағдарламашылары Linux Base стандартымен жүрмеу керек. Бұл бағдарламалық жасақтама қосымшаларына әкелуі мүмкін, бір реттік іске қосуды талап ететіндер, осы тарамда жұмыс істей алмайды, осы тарамда жұмыс істейтін командалық жолақтың аргументтері де, басқа тарамда жұмыс істей алмайды. Бұл пайданушылар мен бағдарламашыларды қатты қынжылтуы мүмкін.



Сурет 8.2 Unix/Linux уақыт графигі (сурет <http://commons.wikimedia.org/wiki/File:Unix.png> берілген, автор белгісіз.)

Бағдарламашы терең айырмашылықты көре алады. Мысалы, Linux ядросы сирек бұрандалы пішінді орындайды, ал Unix ядро ағыны қате болып саналады.

Linux релиздеріне қалай қарайсыздар? 8.3 суретінде көріп тұрғандарыңыздай көптеген әртүрлі пішіндер немесе тарамдар бар, Linux, тағыда әртүрлі «жанама өнім» 20 жыл қарсаңы бойы көп рет шығарылған. Мысалы, суретте көрсетілген SLS / Slackware Linux S.u.S.E Linux әзірлеуде қолданылған, ол өз алдына өзі бетімен OpenSUSE әзірленуінде қолданылды.

Алғашқы Linux ядросының әзірленуі

● Initial development of Linux kernel

● MCC Interim Linux/TAMU Linux (Texas A&M)

● DLD

● SLS/Slackware

● Debian Linux

● Free BSD

○ S.u.S.E.

● Red Hat

○ Knoppix

○ Mandrake

○ Yellow Dog

● Enoch

● Arch linux

○ Oracle Linux

○ Fedora

○ CentOS

○ Morphix

● PC Linux

○ Ubuntu Lite

○ Ubuntu

○ Linux Mint

○ Kanotix

○ Kubuntu

○ Mandriva

○ OpenSUSE

○ Mageia

1991 1993 1995 1997 1999 2001 2003 2005 2007 2009 2011

Сурет 8.3 Linux релиздерінің кілттік тарамдары

Linux негізгі екі бөлінулері Debian және Red Hat болып келеді, Slackware (SLS) қатар, бұлар ең ертедегі дистрибутивтер болған. Бірақ 8.3 суретінде негізгі тарамдар көрсетілген, көрсетілмеген ондаған релиздер бар.

Бәрін тізім шығатындай, тарамдардың өте көп әртүрлі ерекшеліктері бар. Шын мәнінде, сайттар бар, осындай ерекшеліктердің тізімі негізгі себеп болып табылады. Біз мұнда тек төрт шығарылыммен ғана шектелеміз. Red Hat, Debian, Ubuntu и SUSE.

• Red Hat 1994 жылы Боб Янгпен және Марк Юингпен шығарылды, ол әзірлену және қолдану тұрғысынан алғанда Linux нұсқасының алды болып келеді. Оларды кез-келген жерде тізілген құрылғылардан бастап интернет-серверге дейінгі орыннан тауып алуға болады. Red Hat (ОЖ) ашық бастапқы код болып табылады. Дегенмен Red Hat Enterprise Linux

тек ғана сатып алу үшін қол жетімді, бірақ қолдау арқылы қойылады. Red Hat қосымша бағдарламалық жасақтаманы ашық бастапқы код түрінде, қызықты дамуда, қолдайды, өзінің жеке бағдарламалық жасақтамасын иемденіп алып оны ашық бастапқы коды бар қол жетім етті. Кейбір жаңашылдардан Red Hat оларды басқару үшін бағдарламалық жасақтама пакетіне қосады, RPM (Red Hat Package Manager), SELinux (қол жетімді қорған Linux), және де бағдарламалық жасақтама аралық қабат JBoss (БЖ аралық қабаты жанама бағдарламалық жасақтамамен операциялық жүйе арасында мультипроцессерлеуді қабілетті етеді). Red Hat екі жалғасы Fedora және CentOS болып табылады, екеуі алғаш рет 2004жылы шығарылды және тегін ашық бастапқы кодты болды. CentOS Linux қазіргі кезде айтарлықтай веб-сервер үшін тарамды танымал болып табылады.

- Debian 1993 жылы Ян Мердомен шығарылды, Red Hat қарағанда, Debian толығымен коммерциялық емес болу үшін ойластырылған, ұйымдар иемдене алған және қолдау тапқан. Debian 20000 бағдарламалық жасақтамасы барымен мақтана алады. Debian кез-келген Linux үшін аз мөлшерде және тұрақты болған. Debian сонымен қатар, процессорлардың көптеген мөлшерімен жұмыс істей алады, Linux басқа нұсқаларына қарағанда. Операциялық жүйе әзірлеушілеріне шығарылымдар, неғұрлым консервативті болып табылады қолдау ретінде, дегенмен, осы екі артықшылықтары міндетті емес, олар айтылуына қарай жақсы болса да. Әрбір басылым жаңа әзірлемелер мен мүмкіндіктермен шектеледі. Шын мәнінде, Debian негізгі хабарламалары Linux басқа нұсқаларын қарағанда әлдеқайда баяу жылдамдықпен келеді (әр 3 жыл сайын). Debian, Red Hat сияқты ендірілген құрылғылармен жұмыс істей алады. Debian бағдарламалық жасақтама басқаруымен APT (Advanced Package құралы) ұсынылған. Red Hat бастап, Debian операциялық жүйесіне дейін (a Live CD ретінде белгілі) іске қосылу алынбалы бұқаралық ақпарат құралдарынан болуы мүмкін. Мысалы, сіз кез келген Debian CD-ROM немесе USB-құрылғыны жүктеуге болады, оның орнына, Debian, қатты дискке орнатылған. Ubuntu (төменде қараңыз) қатар, Debian бірнеше басқа жалғастырмасы бар, бірінші ретте MEPIS, Linux Mint, и Knoppix, осылардың бәрі жүктелетін дисктен немесе USB құрылғысынан шығарылуы мүмкін.

- Ubuntu 2004 жылы Марк Шаттвортпен шығарылған және Debian тұрақсыз нұсқасынан негізделген, бірақ Debian кетіп қалған, ендеше олар толық байланыста емес. Debian қарағанда, Ubuntu тұрақты шығарылымдарға ие (6 ай), уақыт өткен сайын ұзақ релизді қолдаулар (әрбір 3 жылдан 5 жылға дейін). Ubuntu көп, басқа Linux-релиздеріне қараған-

да, ол жұмыс үстелі үшін тақырыптар шығарады пайдаланушыларға көмек болады, Windows-тан Linux ауысқысы келетіндер үшін.

Қолда бар 3-D әсерлері, және Ubuntu Wiki-стилінде құжаттама үшін тамаша қолдау көрсетеді. Шаттлуорт мультимиллионер болып табылады және бәрімен Ubuntu бөлісті тегін орнатылатын компакт-дискілермен. (бұл тәжірибе 2011жылы тоқтатылды). Debian-ның басқа да басылымдарда ретінде, Ubuntu жүктелетін диск немесе USB-құрылғыдан іске қосуға болады.

- SUSE- Software und System-Entwicklung дегенді білдіреді (бағдарламалық жасақтаманы немісше атауы), ол SuSE болыпта қысқартылады. Slackware, Softlanding Linux Systems (SLS) неміс нұсқасы негізінде Linux нұсқасына негізделген Linux нұсқасы. SLS қайтыс болған сәттен бастап, SuSE содан қолданыстағы SLS ұрпағы болып табылады. SuSE бірінші филиалы 1996 жылы жарияланды және SuSE Linux компаниясы Германияда ірі дистрибьюторы болды. Осыдан кейін көп ұзамай SuSE 1997 жылы, сондай-ақ басқа да елдерде АҚШ-нарығына шықты. 2003 Novell \$ 210 млн SuSE сатып алды. SuSE Linux қарамастан, жоғарыда аталған нұсқалардың, ол әлі Red Hat Пакет менеджері файл құрылымы мен Red Hat тұрады, соның ішінде Red Hat Linux аспектілеріне енгізілген. 2006 жылы, Novell Microsoft корпорациясы келісімге қол қойды, сондықтан SuSE бағдарламалық жасақтаманы Microsoft Windows бағдарламалық құралын қолдай алады. Бұл келісім Free Software Foundation (FSF, келесі бөлімде талқыланады) кейбір дау құрды.

Әліде шатасып тұрсыз ба? Мүмкін солай.

АШЫҚ БАСТАПҚЫ КОДТЫҢ АЛҒА БАСУЫ

Linux даму Unix тарихында соңғы іс-шара болып табылады. Бұрын, Unix жұмыс істегендер екі түрлі жолмен жүрді. Бір жағынан, қорғаныс Advanced зерттеу агенттігі даму (DARPA) қаржыландыратын жұмыстан Беркли университетінде әзірленген BSD Unix, нұсқасы мен жұмыс істейтін адамдар болған. BSD әлемнің түкпір-түкпіріде пайдаланушылар бар, бірақ негізінен батыс жағалауы хакерлер тобы болды, дұрыстуға көмектесті, қолдауға және жақсартуға Unix. Соңында 1970 немесе 1980 жылдардың басында Шамамен сол уақытта, Ричард Столлмен, Массачусетс технологиялық институтында жасанды интеллект зерттеушісі, GNU * ретінде белгіліөз жобасын әзірледі. GNU Столлмен бағдарламалық жасақтамасы тегін болды, және құны тұрғысынан емес, сонымен қатар болашақта бағдарламалық жасақтаманы дамыту үшін

адамдардың қабілетін тұрғысынан ғана емес, еді, мұндай Unix операциялық жүйесі үшін бағдарламалық жасақтама дамыту болып табылады. Столмен көзқарасы еркін бағдарламалық жасақтама қозғалысқа айналды. Бұл идея қозғалыс бөлігі ретінде дамыған кез келген бағдарламалық қоғамдық қазына болады, және адамдар оны өзгерту үшін тегін қолданды. Осылайша, бағдарламалық жасақтама код тұрғысынан (соңды ол болған жағдайда, сирек болатын нәрсе) қол жетімді болуы тиіс. Бастапқы код өзгерді, содан кейін қайтып жобасына жаңа код үлесін қосуға қабілетті болуы мүмкін. Бұл топ өзі GNU деп аталатын, Unix-тәрізді операциялық жүйені (немесе кейде GNU / Linux) дамытуға тырысқан Unix пайдаланушылары. Атап кеткен жөн, Столмен жобасы 1983 жылы басталды, ал GNU тұрақты нұсқасын ешқашан шығармаған.

* GNU Unix емес GNU деп шифрланады. Бұл рекурсивті анықтау, бұл шын мәнінде олай емес. Бұны Столлман беруге тырысты, яғни операциялық жүйе емес, Unix-ұқсас операциялық жүйеден тұратын тек еркін бағдарламалық жасақтама болды.

1985 жылы, Столмен FSF қалаған. FSF, және еркін бағдарламалық жасақтама қозғалысының бүкіл ұғымдар, комбайн және социологиялық және саяси көзқарастарды – пайдаланушылар өзгертті, сондай-ақ бар немесе өзгеріссіз көшірмелерін қайта бөлуге, оны оқуға, оны іске қосу үшін бағдарламалық жасақтама тегін қолжетімді болуы тиіс.

Бұл Open Source бастамасы (OSI), ашық бастапқы бағдарламалық құралмен бағдарламалық жасақтаманы қолдап, 1998 жылы құрылған ұйым бағдарламалық жасақтаманың еркін қозғалысы болып табылады. Алайда, OSI FSF бірдей емес; шын мәнінде, (негізінен Столлман атайды туындаған) олардың арасында елеулі айырмашылықтар бар. Еркін бағдарламалық жасақтама қозғалысының тұжырымдамасына сәйкес, барлық бағдарламалық жасақтама тегін болуы тиіс. Бағдарламалық жасақтаманы сату этикалық және моральдық жағына дұрыс болып саналмады. Бағдарламалық жасақтама өзі идеяларды жүзеге асыра алатын болып табылады, және идеялар барлық қол жетімді болуы тиіс; Осылайша, бағдарламалық жасақтам барлығы үшін қолжетімді болуы тиіс. Мысалы, OSI тартылған немесе қосылғандар, осы көрініс тым шектеу болып табылады. Біреудің мансабы бағдарламалық жасақтаманы құру болып табылады, онда, әрине, оған ақша табу үшін құқығы бар. Бағдарламалық жасақтаның еркін қозғалысы орнына заңсыз авторлық құқықпен қорғалған бағдарламалық жасақтаманы көшіруге және пайдалануға, олар хакерлерден сақтауға болатын болды.

Бағдарламалық жасақтама көзі ашық қоғамдастыққа үлес қосты, және ашық қалуы тиіс, таза коммерциялық бағдарламалық жасақтама болуы керек, (мысалы, Microsoft) шектен және таза еркін бағдарламалық жасақтама (FSF) арасындағы Қауымдастық көзі ашық болған. Дегенмен, коммерциялық өнімдерді дамыту үшін ашық бастапқы бағдарламалық жасақтаманы пайдаланады, сондай-ақ ұзақ басқалар жасаған жарналар әлі еркін қол жетімді ретінде оның құқық және сату еркін көшірмесін қолжетімді болуы тиіс. Бұл GNU General Public License (GPL) дамуына әкелді. GPL GNU жобасы Столлман айтқандай жазған болса да, ол шын мәнінде тұрақты компания үшін пайдаланыланылған. Бұл бағдарламалық жасақтама бөлу және модификациялау үшін еркін қол жетімді дейді, әлі GPL лицензиясы өзгертілген ештеңе сақтайды. Столлмен авторлық құқық болмаған, авторлықтың болмауы GPL деп аталады. GPL, Linux, болғанға әсер етуіне мүмкіндік берді. Шамамен 50% Red Hat Linux 7.1 Linux орталарға жазбаша бағдарламалық жасақтама көптеген сияқты, оған GPL кіреді. Шын мәнінде, GPL барлық еркін бағдарламалық жасақтама жобаларының шамамен 65% -ды құрады. *

Бүгін ашық көзді қоғамдастық өркендеп келе жатыр. Мындаған адамдар бағдарламалық жасақтамада жұмыс істеуге және жаңа идеяларды, бағдарламалық жасақтама, жаңа мүмкіндіктерді, сондай-ақ жаңа бағдарламалық өнімдерді алуға бару үшін уақытқа бағынышты болды. Linux жаңа кодына жарналар құрайды, ол ешқандай әлемдік қоғам болмады, өйткені жасағаны табыстарға жететініне күмән келтірген. Қауымдастық бастапқы ашық кодты Linux ғана ықпал жасаған емес, сонымен Linux ірі жоба және ең елеулі жетістік қауымдастығы болып табылады. Ол сондай-ақ компаниялардың кейбір олардың іскерлік тәсілдерін қарау фактісі әкелді. Мысалы, Microsoft өзі ашық бастапқы кодты бірқатар жобаларды бар.

WINDOWS ТАРИХЫ

Windows тарихы, сезімі дербес компьютерлер тарихын, сондай-ақ Windows көрсетеді, және оның ізашары, MS-DOS, мұнда талқыланды жеке * GPL лицензия, ең танымал операциялық жүйе, шын мәнінде, екінші және үшінші нұсқалары, GPLv2 және GPLv3 табылады болды тиісінше, 1991 және 2005 жылы шығарылған.

Операциялық жүйе тарихы ■ 227

Алайда, толық Windows тарихын түсіну үшін, біз сондай-ақ әр түрлі амалдық жүйені, Macintosh қарастыру қажет. Алдымен, біз MS-DOS қа-

растырамыз.

MS DOS операциялық жүйесі 1981 ж IBM дербес компьютерімен бір мезгілде пайда болып, әрбір компьютерге орналастырылған болатын. IBM ашық сәулет пайдалануға, оның бірінші дербес компьютер нарығы дизайн, деп шешім қабылдады; олар Intel 8088 микропроцессоры, соның ішінде компьютерді құруға дайын барлық компоненттерді пайдаланылған. Бұл шын мәнінде, IBM PC (немесе баламасы) типі болып табылады, көшіру және босату үшін басқа компанияларға ұсынылатын болып табылады. Бұл компьютерлер бірдей процессор болуы болғандықтан, олар бір бағдарламамен жұмыс істей алатын. Осылайша, MS-DOS қолжетімді ДК IBM бар, ол сондай-ақ ДК клондарымен жұмыс істейтін болады. (IBM қоспағанда) компанияның дамуына көп инвестиция керек, және (2) платформасына арналған қосымша бағдарламалық құрал өндірілген жоқ, сондай-ақ (1) Компьютерлер, оның көптеген бәсекелестеріне қарағанда арзан болған, өйткені IBM PC клондарының танымалдығы негізінен болды, нарықта осы түрдегі көп компьютерлер болды. Олардың танымал ретінде өсіп, нарықта MS-DOS үлесі.

БІРНЕШЕ ОПЕРАЦИЯЛЫҚ ЖҮЙЕЛЕРДІ ҮЙРЕНУ

АТ кәсіби маман ретінде, сіз бірнеше операциялық жүйелерді білуіңіз тиіс. Алайда, сіз осы тапсырманы орындау үшін көп компьютерлерді сатып алу қажет емес. Мұнда біз Сіздің компьютеріңізде бірнеше жүйелердің қатысуымен үш тәсілдерді қарап шығамыз. Біріншісі қарапайым болып табылады. Егер сіз OS X нұсқасын іске қосатын Apple Macintosh бар болса, терезе XTerm ашу арқылы сіз кез келген уақытта UNIX алуға болады. Ол утилиталармен қол жетімді. Терезе Xterm FreeBSD және NetBSD UNIX нұсқаларында кем дегенде бір бөлігіне негізделген Mach ядро бар. Бұл, шамамен айтқанда, UNIX емес болып табылады, бірақ ол іс жүзінде барлық жағынан өте ұқсас көрінеді. Қандайда бір компьютер тұғырнамасын пайдаланғаныңызға қарамаст, тағы бір тәсіл, виртуалды машинаның орнату. Виртуалды машина көптеген машиналар түрлерін эмуляциялай алады. Осылайша, егер сіз компьютердегі В.М. орнатқан қалыпты режимде компьютерді бастаңыз содан кейін сіздің виртуалды машинаға жүктелген бағдарламалық жасақтама В.М., іске қосыңыз. Виртуалды машинасы кейіннен сіз бірнеше түрлі операциялық жүйелер іске қосуға аласыз. Осы тәсіл кемшіліктері (Linux-ақ, әдетте, тегін, Windows және Mac мұндай емес, ал) сіз сатып алынатын өнімді талап ете алады, ол операциялық жүйені орнату қажет болады деп табылатындығында. Сонымен қатар, виртуалды машина тиімді жұмыс істеу үшін

ресурстарды көп алады, сондықтан сіз виртуалды машинасын бастаған кезде сіздің жүйесіңіз бәсеңдемейтін деп табуыңыз мүмкін. Ядролы процессор бұл мәселені жояды.

Егер сіз Windows бар машинаны пайдаланып жатсаңыз және сіз виртуалды машинасын пайдалану қажет емес болса, сіз қос жүктеу үшін шектелесіз. Қос-қосылғаннан компьютер – екі операциялық жүйелер (немесе одан да көп) қолжетімді. Компьютеріңізді бастағанда, Windows жүктеушісі тоқтатыла тұруы мүмкін және сіз Linux жүктеушісін бақылау аласыз. Екі ең танымал жүктеу тиегіш LILO (Linux Loader) және GRUB (Гранд бірыңғай тиегіш) болып табылады. Осылайша, Windows және Linux сіздің қатты диск кеңістігіңізге ортақ, және, әдепкі Windows жүктеу үшін орнатылса, дегенмен, сіз кез келген уақытта Linux операциялық жүйесін іске қоса аласыз. Macintosh, немесе виртуалды машина көзқарас айырмашылығы, алайда, сіз ашу операциялық жүйені өшіріуіңіз керек болады.

DOS бір-мақсатты, мәтін (емес GUI) операциялық жүйе болды. Командалар негізінен файлдық жүйеде жұмыс істейтін болды: тізімделер өзгерту, тізімделерді құруға, файлдарды жылжыту, файлдарды көшіру, файлдарды жою, сондай-ақ іске қосу / бақылау бағдарламалары. Бір-мақсатты операциялық жүйе ретінде, операциялық жүйе пайдаланушыны шақыру үшін экранды қалдырып, жұмыс істейтін болады. Пайдаланушы көрсетілген команданы енгізу керек. Операциялық жүйе командасын орындайды.

Пайдаланушының командасы бағдарламаны іске қосу керек болса, бағдарлама сондай-ақ ұзақ жерде аяқталып, пайдаланушыға басқару үшін қайтару мүмкін емес болады, себебі жұмыс істейтін болады. Бағдарлама кейбір кіріс / шығысты талап жағдайында осы бір мақсатты сипаттағы ғана ерекшелік, бұл жағдайда, бағдарлама операциялық жүйесінің пайдасына тоқтатылады, немесе пайдаланушы Ctrl + S немесе CNTRL + Alt + Del теру арқылы, мысалы, үзу нәрсе жасайды. Ең бағдарламалық жасақтама, сондай-ақ, көрсеткі пернелерін және пернелер немесе функционалдық пернелердің белгілі комбинацияларды пайдалану арқылы қол жеткізуге болады (мәтін) мәзірді пайдаланылатын көптеген бағдарламалық өнімдер, дегенмен, мәтін негізінде болады.

1970 жылы Xerox Palo Alto (Xerox Parc) ғалымдары өздерінің ОЖ үшін әзірленген GUI болды. Олар назарға коммерциялық өнімге әкеп соғатын жетуді есепке алмай, негізінен жасанды интеллект саласындағы өз ғылыми-зерттеу жеңілдету мақсатында жасаған. Стив Джобс және Стив

Возняк, Apple Computer Xerox PARC барып, терезе операциялық жүйесін айтарлықтай әлі қанаты қатая қоймаған дербес компьютер нарығы әсер етуі мүмкін екенін түсіндірген.

Apple, кейін көп ұзамай жаңа операциялық жүйенің дамуын бастады. Стив Джобс графикалық интерфейсі бар бірінші дербес компьютер болу керек еді деп Apple Lisa, жұмыс істеуге барды. Алайда, жобаның шығындарының қалып, жаңа жұмыс орындарын жобаның аумағынан аластатылды. Ол содан кейін Apple Macintosh тобына қосылды. Лиза Macintosh асып және бірінші болғанымен, ол өте нашар сатылды ішінара өте жоғары құнының (бірлігіне \$ 10 000) байланысты. Macintosh бір жыл өткен соң сатылған және бірден ұстап болады. Apple Macintosh амалдық жүйесі Mac табысқа жету үшін алғашқы коммерциялық негізіндегі графикалық операциялық жүйе интерфейсі болып саналады. Macintosh дамыту фазасы бойынша, Apple қолданбалы бағдарламалық жасақтама дамытуға көмектесу үшін, Microsoft жалдағанады. MS-DOS дамытумен айналысады, Билл Гейтс, Windows және Microsoft корпорациясы мән ұқсас өнімді дами бастады түсіндірді.

MS-DOS 1981 жылы шығарылғанына қарамастан, ол 1970 жылы ең әзірленуде болды. 1970 жылдың екінші жартысында, Microsoft бастапқыда интерфейс менеджері деп аталады, Windows жұмыс істей бастады. Ол алдымен MS-DOS бар компьютерді бастау керек болды, содан кейін сіз оның үстіне GUI іске қосу дегенді білдіреді, MS-DOS жоғарғы жағында орналастырылатын болды. Себебі MS-DOS оған тәуелді, осы пайдаланушы іс жүзінде GUI пайдалануға немесе қажет болған жағдайда DOS командалық жолында оралуға болады дегенді білдіреді. Windows 1.0 1 жыл Apple Macintosh шығарғаннан кейін, 1985 жылы жарыққа шықты. Macintosh барлық ескерген кезінде ешқандай сызықтар болған; Егер сіз графикалық интерфейстан бастап бәрін іске қосу керексіз.

Осы сәтте, Microsoft, 1987 жылы Windows 2.0 Windows 1.0, және 1990 жылы Windows 3.0 орындаңыз, Windows Intel процессор 80386 болып (немесе 386) іске қосылу үшін әзірленді. 1992 жылы, Windows 3.1 шығарылды, және сол уақытта, ең танымал және кеңінен барлық операциялық жүйелердің пайдаланылатын болды, ең алдымен, Intel 486 процессоры бағытталған. Барлық осы OS MS-DOS салынған, және олар мультибағдарламанландыру жүйелерінде барлық пайдаланушы мәжбүрлеп бір үрдіске (Windows) ауысуға болады. Ол кооперативті көп міндетті нысан болып табылады. (Windows, 2.1 нұсқасы қоспағанда арнайы аппараттық және бағдарламалық бастап) ешнәрсе шынайы (конкурстық) тапсырмалылықты орындауға болмайды.

Шамамен сол уақытта, сондай-ақ, Microsoft Windows NT, 32 биттік,

желілік операциялық жүйені шығарды. Ол өзінің 32-биттік сипатына тезірек байланысты болғанын, Windows NT клиент-сервер желілер үшін арналған. Бұл бәсекеге қабілетті тапсырмалылықты қолдау мақсатында, сондай-ақ, Windows алғашқы нұсқасы болды. Windows NT ұйымдастыру бойынша айтарлықтай әсер жалғастырды.

Windows тарихында 1 дамытудағы келесі қадам, Windows 95 шығарылымы болды. Windows 95, Windows NT-ден Windows 3.1 кейбір мүмкіндіктерін үйлестіре отырып, кіріктірілген Интернет мүмкіндіктері бар бірінші операциялық жүйе болды. Windows NT сияқты, сондай-ақ, Windows 95 бәсекеге қабілетті мульти мақсатты болды. Windows 95 тағы бір жаңа мүмкіндік дайындығын жасауға қабілеті болды. Пайдалануға дайын пайдаланушы оңай жазға жаңа жабдықтар орнатуға болады дегенді білдіреді. Бұл олар қосылады, компьютерлік құрылғыны тани алады дегенді білдіреді. Windows 95 Intel Pentium процессорлар арналған. Жылдам, Windows бірнеше жаңа нұсқаларында, 2001 жылы Windows XP (NT-негізделген нұсқасы) және 2003 (Windows Server), (Windows 98), 2000 [Windows 2000 және Windows Millennium Edition (Windows ME)] 1998 жылы шығарылды. Сонымен қатар, 1996 жылы, Microsoft Windows CE, ұялы телефондар және басқа да құрылғылар «төмендеді» (мысалы, навигациялық құрылғылар) олар үшін Windows өз нұсқасын шығарды. 2007 жылы, Microsoft жаңа буын терезе жүйесін, Vista шығарды. Vista, жаңа қауіпсіздік мүмкіндіктері бар бірқатар болғанымен, ол төмен өнімділігі мәселелерін тап болды. Ақыр аяғында, 2009 жылы, Windows 7 шығарылды.

Тағдыр қызықты бағыт жылы, өйткені нарықта Microsoft корпорациясы ірі үлеске ие болды, Apple компаниясының бизнес боп қалануына өзінің ең жақсы тырысты. 1997 жылы, Microsoft Apple \$ 150 млн инвестициялады. Нәтижесінде, ол Apple Computers айтарлықтай әсер етті. Біріншіден, 2005 жылы, Apple Macintosh Intel процессор RISC негізделген ДК қуат процессоры қосулы. Екіншіден, өйткені ол Macintosh компьютерлер Windows жұмыс істей алады бағдарламалық жасақтаманы іске асыруға мүмкіндік беретін өзгерістер болды. Macintosh сіздің қазіргі X нұсқасыңыз, Mac ретінде белгілі, олардың Windows операциялық жүйесіне негізделген сонда сақталады. Бастапқыда MS-DOS Windows негізделген болатын, айырмашылығы, Mac OS X Unix операциялық жүйесінің жоғарғы жағында болатын. Біреу алғысы келсе, сіз Unix қоршам ашу және пәрмен жолағында жұмыс істей аласыз. Windows әлі MS-DOS бар, сондықтан сіз DOS жолын ашуға болады.

9. Bash қабықшасы және түзету.

Компьютердің басым қолданушылары мен ИТдің бастапқы студенттері үшін Linux жаңа тәжірибе болады. Негізінде, көбіне командалық жолақ таныс емес территория болып табылады. Бұл бөлім Bash қабықшасына арналып, Linuxтің барлық дистрибутивтарында қолданылады. Көбінесе, бұл бөлім Bash қабықшасы командалық жолақтың өзара әрекеттестігін ұсынатын ерекшеліктердің қатарын қамтиды. Тарихи тізімі және лақап аттардың қолданылуы арқылы тақырыптар командалық жолақтың түзетуін енгізеді. Сонымен қатар, түсіндіруші Bash зерттелінеді.

Бөлім екі текстік редакторлардың VI және Emacs қарастырылуымен аяқталады.

Бұл бөлімнің оқыту мақсаты:

- Командалық жолақтың енгізуін жеңілдететін Bash жарлықтарына кіріспе
- Bash түсіндірушісінің сипаттау рөлі.
- Linux командасына қайта бағыттау кіріспесі.
- Bash қабықшасының механизмдерін дербестендіруі үшін кіріспе.
- Linux текстік редакторының маңызды сипаттамасын талқылау
- VI және Emacs редакторларының қолданылуы ұсынысының құрылымы.

Бұл бөлімде Linux нақты тақырыптарына тоқталамыз. Біріншіден, біз Bash қабықшасын толығырақ қарастырамыз. Біз командалық жолақтағы жазудың түзетуіне, ауыспалылығына, лақап аттарына, және Bash қабықшасын дербестендіруін қарастырамыз. Біз қарасырып отырған міндеттер графикалық қолданушының интерфейсі емес (GUI), Linux қолданушыларын командалық жолақта қолданылуын қолдайды. Біз әлемде қолданылып жатқан Unix / Linux: VI және Emacsпен қатар екі жалпы бағдарламалық текстік бөлімін аяқтаймыз. Emacs-те басылып жатқан пернетақталардың командалық жолақтары да түзету команда-

сындағыдай, бірақ бұл редакторлық тақырыптағы Bash пен мүлдем байланыспаған болып көрінуі мүмкін.

Қабықшалар

Қабықша түсіндіруші ортасы болып табылады. Бұл дегеніміз сіз командаларды енгізіп, пернетақтаның басылуы кезінде бұл командалар түсіндіріледі. Командалардың түсіндірушісі команданың кіру нысанынан орындалу нысанына ауысуын талап етеді. Түсіндірушілік бағдарламасы аударылады және жолақтан жолаққа орындалады, сол уақытта бағдарламаның дәстүрлі құрастырылуы толықтай аударылады, кейін орындалады.

Linux жолақтарының командалық түзетулерінің қиындықтарынан біз өздеріне көп командаларды қосатын, кіргізуі мен шығарылуының ауысымы, сонымен қатар көбіне құрылымында қиын жағдайларға тап болатын қысқартылған операцияларда Linuxтің күрделі командаларын енгізе аламыз.

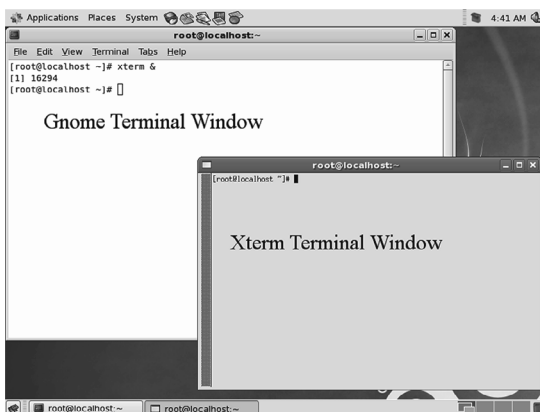
Linux / Unix қабықшасы мамандандырылған болып табылады. Қолданушылар лақап аттарды айтатын, сол сияқты~ (қолданушының үй жайындағы тізімін көсететін), қысқартылған операцияларды қолдану, тез арада орындалатын операцияларды қоса алатын, командаларда қолданылатын және қолданушының ішкі сипатын кіргізетін ауыспалы ұғымдарды анықтайды. Қоршаған ортаның бейімделіуін командалық жолақтың және әртүрлі файлдардың түзету жолы арқылы іске асады. Келесі бөлімде көретініміздей, бейімделе алатындай әртүрлі заттардың жалпы қатары бар.

Сонымен қатар қабықша басты мәтіннің негізінде, қабықша әрекеттестігі күрделі тапсырма болуы мүмкін. Қолданушылардың көбісі бұл (ОЖ) операциялық жүйенің командалық базасымен таныспаған, себебі олар тек операциялық жүйемен тек графикалық интерфейс арқылы ғана жұмыс істейді. Басқа жағынан қарағанда, қабықша дәрежесінің өзара әрекеттестігінде иілгіштік және бақылау дәрежесін қамтамасыз етіп қана қоймай, себебі қабықша өзі талап етпейдіктен, ол ресурстарды үнемдейді, мысалы, күрделі графикалық командаларды және тінтуірдің қадағалауы. Қабықша бұл –компьютерге қосылудың жеңіл түрі болып табылады, бірақ сіз GUI қабатынан бас тартқандықтан, компьютермен төменгі деңгейде сөйлесе аласыз.

Егер сіз қолданушының графикалық интерфейсі өшірілгенде сіз компьютерді іске қосуға мүмкіндік алып, Linux / Unix қабықшасы автоматты түрде ашылады. Бұл, мысалы, Telnet және SSHті компьютерді

қосу кезінде қолданғандағы жағдай.Сіз сонымен қатар, терминал терезесіндегі бағдарламаны іске қосып, қапшықты аша аласыз. Мысалы, CentOSты, сіз тінтуірдің оң жақ батырмасын басқан кезде жұмыс үстелін немесе терминалдың қосымшалар терезесін таңдап, терминал терезесін ашасыз. Сіз, сонымен қатар, командалық жолақта мысалы, гном-терминал командалары жолымен, Gnome терминал тезесін ашу үшін, немесе Xtermді теріп, жаңа X тезесін бастау үшін терминалдың жаңа терезесін аша аласыз.X терезелік жүйесі Gnomедан үлкен. Xterm терезесі жойылған кезде, бірқатар мәзірі болғандықтан Gnome терминалының терезесі графикалық интерфейспен әрекеттесуіне мүмкіндік болады. (Сурет, 9.1). Қолданушы бірнеше терминалдық терезені аша алады, әрбірі оның жеке қабықша түсіндіруші ортасы болып келеді. Қабықша үнсіздікте болса да, қолданушының қабықша логині арқылы, сіз әрбір терезеде әртүрлі қабықша аша аласыз. Солай-ақ, сіз қабықша ішінің жұмысы арқылы жаңа қабықша сеансін бастайсыз. Бұндай сәтте жаңа қабықша (қабықшаның ішкі сеансын қалай атау мүмкіндігіне ие болдық) алдын ала берілген нәрсені қайта жауып тастайды. Ішкі қабықшадан шығар алдында, сіз өзіңіздің алдыңғы сессияңызды қайтадан ретке келтіре аласыз.

Бірінші рет 1977 жылы шығарылған,Unix қабықшасы ерте кездегі қабықшаның бірі Bourne Shell (sh) деп аталды. Кейінірек қабықшыларғv C (CSH) қабықшасы, Tenex C сондай-ақ T (Tcsh) қабықшасы, Korn (ksh) және Almquist (ash) қосылды. Бұдан басқа бір қатарлары бар, аса атақты немесе аз қолданылатындар Busybox, Hamilton C қабықшасы, fish, Perl және zoidberg қабықшасы.



Сурет 9.1 Терминалдық терезелер

Sh, CSH, Tcsh, ksh және ash атаулары қабықша бағдарламасыны нақты атауы болып келеді. Мысалы, командалық жолақта CSH командасын теру арқылы, C қабықшасын іске қосуға болады. Келесі азат жолда жазылғандай, бұл жаңа қабықша сессиясын құрады. Bourne қабықшасы негізінде ,Bash (bash) қабықшасы іс жүзінде Bourne Again Shell атауымен атақты. Біз оны жай Bash деп атайық. Келесі 3 бөлімде біз қабықшаны толығырақ қарастырамыз. Айта кетейік, жоғарыда аталып кеткен қабықшаларда және Bashта көптеген функциялар қол жетімді.

BASH ҚАБЫҚШАСЫНЫҢ ТҮЗЕТУ ФУНКЦИЯСЫ

Bash қабықшасы Bourne қабықшасын сапасын жақсарту мақсатында ауыстырылды.

GNU ОС қадағалау үшін Bash жазылды, бұл әрекет Unix операциялық жүйесін құру үшін ашылған алғашқы код болатын. Кейінірек бұл Linux және ОС Mac X үшін құрылған унсіздік қабықшасы болатын. Bourne қабықшасы сияқты, Bashта көптеген өзіндік функцияларды атқарады, бірақ CSH и KSH қабықшалары үшін арналған функцияларды да атқарады. Бұл солай-ақ, Bourne қабықшасындағы қажетті синтаксис бөлшектерінің жұмысын жеңілдетеді, бұған қарамастан қабықша үлкен болғанда, оны қолдану жеңілдірек болады.

Бұл бөлімде біз қосу және түзету командасын жеңілдететін, Bashқа қатысты бірнеше функцияны қарастырамыз. Бұлар

- Тарих
- Командалық жолақтағы түзету
- Табтын аяқталуы
- Лақап аттар(alias)
- Ашық жақша
- Тильдалар
- Нышан

Келесі бөлімде Bash қабықшасының басқада ерекшеліктерін қарастырамыз.

Bash сеансында әдеттегі командасын қосқанда, әр жолы ол тізімдер тарихы деген атауға көшіріп алады. Сіз бұрын енгізілген команданы тез табасыз, себебі ол тізімдер тарихында көрсетілген. History командасын теру арқылы тізімдер тарихын қарай аласыз. Сурет 9.2 суретінде тізімдер тарихының үлгісі көрсетілген. Ол пайдалы және танымал болғаны соншалық, сол уақыттан бастап көптеген қабықшаларға енгізіле баста-

ды, тарихта алғаш рет CSHге қосылған.

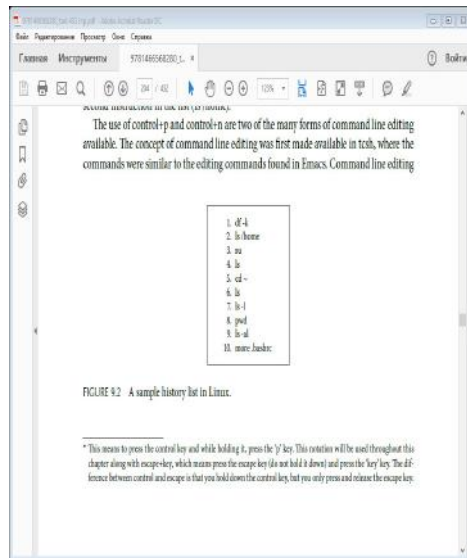
Тізімдер тарихынан еске түсірудің бірнеше командалық тәсілі бар. Командалық жолақтан `ctrl + p *` немесе жоғары нұсқаулығы бар пернені теру арқылы алдыңғы команданы шығаруға болады. `Ctrl + p` немесе жоғары командасын басуды жалғастыру арқылы тізімдер тарихынан соңғы командаларды көре аласыз. Мысалы, егер сіз 9.2 суретінде көрсетілгендей `CNTRL + P`ді 3 рет басып 10 команда енгізсеңіз, "PWD"ті командалық жолақта қалдырасыз. Enter басып команданы орындаңыз. Бір қадам артқа жылжып, жоғары нұсқаулығы бар пернені немесе `ctrl + n` (`p` алдыңғыға өту үшін, `n` келесіге) басу арқылы тізімдер тарихы бойынша алға жылжисыз.

Тізімдер тарихын еске түсірудің басқада командалары ол пернетақтада `!number`ді теріп жазып, `number` тұрған жер тізімдер тарихының жазуының номері болып саналады (көбінесе леп белгісін жарылыс деп атайды). Мысалы, 9.2 суретінде көрсетілгендей `"ls / home"` командасын `!2` орындайды. Соңғы команданы қайталау үшін, `!!` (ПИФ-паф) енгізіңіз. Сіз солай-ақ мынаны қолдана аласыз! Сіз `!chars` басу арқылы, `chars` командасы сіз қайтадан көре алатын біріншінышан болып табылады.

9.2 суретінде көрсетілгендей 11 командасы бойынша тізімдер тарихы `sr` `* / home / zap-paf` болып табылады. Содан соң `!` енгізсеңіз, онда соңғы 'C' басталатын командаларды көрсетеді, `sr`. Командасы. Егер `!cd` енгізсеңіз, 5 нұсқаулық болады, онда ол 'CD' басталатын соңғы командаларды көрсетеді, `CD ~`. Түп тура осылай `!!` (кіші әріппен 'L') 9 нұсқаулықты қайталайды (`ls -al`), `!!s` және `!!s` бұны қалай істейді (`Ls` дефисынан кейін кеңістікті жалғастыру арқылы). Егер біз `!!s/ны` енгізсек, онда ол (`ls / home`) тізімінен екінші команданы қайталайды.

Командалық жолақта `ctrl + p` және `ctrl + n` көптің ішінен екеуін қолдану түзету нысанының қолайлысы. `Emacs`ті ойлап тапқанда, командалар түзету командаларына ұксас болғанда, түзету тұжырымдамасы командалық жолақта `Tesh` бірінші рет ұсынған.

Командалық жолақтың түзетілуі



Сурет 9.2Linuxтағы тізім тарихының үлгісі

* бұл дегеніміз, басқару пернесін басып тұрып 'P' пернесін басу. Бұл белгі бастаптық жүктелімде escape + key пернесін басып, бұл дегеніміз Escape пернесін (оны басып тұрмай) және "key" басу. Ctrl және escape пернесін айырмашылықтары, ctrl пернесін басып ұстап тұрасыз, ал escape пернесін басасыз және жібересіз.

Қазіргі кезде командалық жолақта тұрған текстте іске асырылады. Сіз командалық жолақта мензермен орнын ауыстыруды, нышандарды қиып алу және/немесе нышандарды қою, сондай-ақ ctrl немесе escape және де басқа да пернелерді қиыстыра қолдануға болады. Төменде түзету операциясының аса пайдалы тізімі көрсетілген. c + char немесе m + char дегеніміз, c + char - нышанды басуды және ctrl пернесін басып тұруды білдіреді (мысалы, c + n "ctrl пернесін басыңыз және ұстап тұрып N басыңыз"), және m + char—нышаннан соң escape пернесін бас керек (мысалы, m + f дегеніміз "escape пернесін басып, содан соң F пернесін басыңыз"). Не себепті біз escape үшін 'm' қолданамыз, өйткені көп адамдар escape пернесіне "meta" пернесіне қарағандай болады.

• c + n – командалық жолақтағы тізімдер тарихынан келесі команданы

көрсетеді

- $c + p$ - командалық жолақтағы тізімдер тарихының алдыңғы команда-сын көрсету
- $c + f$ - меңзерді бір нышан жоғары ауыстыру
- $c + b$ - меңзерді бір нышан кері ауыстыру
- $m + f$ - меңзерді бір сөзге жоғары ауыстыру
- $m + b$ - меңзерді бір сөзге төмен ауыстыру
- $c + a$ - меңзерді жолдың басына ауыстыру
- $c + e$ - меңзерді жолдың соңына ауыстыру
- $c + d$ - меңзер тұрған жердегі нышанды өшіру
- $c + k$ - мазмұнды меңзерден бастап жолдың соңына дейін өшіру
- $c + w$ - сөздің басында меңзердің алдында тұрған барлық нышанды өшіру
- $c + u$ - жақында өшірілген барлығын алу ($C + W$ немесе $C + k$ көмегімен) және кері қайтару, ағымдағы меңзер позициясын орналастыру(бұл кірістіру операция)
- $c + _$ - соңғы түзеті операцияны жою (өшіру немесе орналастыру)

Әрине сіз алдындағы нышанды өшіру үшін, Backspace қолдана аласыз, del өшіру нышанын, алға немесе артқа ауыстыру үшін, нұсқаулығы солға және оңға қарайтын пернені, сондай-ақ жолдың басына және де соңына өту үшін home немесе end пернесін қолданылады.

Түзету пернелерінің байланысуын жаттағаннан кейін, командалық жолақтағы команданы қосу кішкене де болса жеңілiрек болады. Мысалы, сіз бес файлды ағымдағы тізімдемеден тізімдемеге көшіргіңіз келеді / home/zappaf, әр файлдың атауы өзгереді. Бес файл file1.txt, file2.txt және т.б., деп аталады, сіз оларды zappa1.txt, zappa2.txt және т.б. өзгеркіңіз келеді. Бірінші команда түзетуді қолданбай енгізіледі.

file1.txt /home/zappaf/zappa1.txt <enter> жол

Қалған төрт команда түзету жолы арқылы жеңіл іске асырылады. Команданы алу үшін $c + p$ басыңыз. Жолдың басына өту үшін, $c + a$ басыңыз. $m + f$ екі рет басыңыз.

Екі сөзге ілгері жылжу. Меңзер енді file1 және TXT арасында. 2нi енгізу үшін, 1дi backspace өшіріп 2 енгізіңіз. Жолдың соңына бар үшін, $c + e$ басыңыз. “TXT” в zappa1.txt басына ауыстыру үшін, $m + b$ басыңыз. Нүктеге өту үшін, $c + b$ басыңыз, 1 немесе 2 орналастыру үшін Backspace-пен өшіру. Енді команда:

file2.txt /home/zappaf/zappa2.txt жол

Енді өзгертілген команданы қосу үшін <enter> басыңыз. Бұл нұсқаулықты басқа да файлдарға қолдана алатындай, өзіңіздің қадамыңызды қайталап отырыңыз.

Bash-тың тағы да басқа аса пайдалы функциясы автотолтыру деп аталады. Егер сіз бірегей атау қою бөліміне дейін келсеңіз, тізімдеме немесе файл атауларын қосуда, табулия пернесін басқан кезде түсіндіруші атауды аяқтайды. Мысалы, қарастырайық жұмыс тізімдемесінен келесі файлды тауып алдыңыз:

forgotten.txt frank.txt fresh.txt functions.txt funny.txt
other_stuff.txt

Сіз ішінен аз қолданылатын файлды көргіңіз келеді. *Егер сіз less fo <Tab> терсеңіз, тек forgotten.txt "fo"дан басталғандықтан Bash түсіндірушісі тауып сіз үшін аяқтайды. Егер оның орнына less fr <Tab> терсеңіз, онда Bash оны атқара алмайды, себебі сіз frank.txt немесе fresh.txt қайсысын таңдарыңыз айқын болмайды. Ондай жағдай да, түсіндіруші дыбыстықсигнал береді, өйткені файл бірегей емес. Егер сіз less fra <Tab> терсеңіз, ол frank.txt болу үшін аяқталады, егер сіз less fre <Tab> терсеңіз, ол мүмкін fresh.txt болып аяқталуы мүмкін. Барлық файл атауларын түгендеу үшін, tab екі рет басу арқылы Bash түсіндірушісін шақырамыз (немесе егер көп файл болса, онда сіз хабарлама аласыз, немесе барлық файл белгілену керек). Сондай-ақ, Bash frank.txt и fresh.txt тізімін ашу үшін, less fr <Tab><Tab> қосады.

Linux-та, alias тез іске қосу командасы болып табылады. Сіз өз alias енгізгенде, ол толық командамен өзгереді, сондықтан лақап ат тандаңыз. Кез-келген уақытта командалық жолақта alias қоюға болады, немесе олар басқа файлдарда анықталуы мүмкін (біз Bash дара қабықшасында осы файлдардың қолданылуын қарастырамыз). Alias кескінін айқындау

alias name = командасы

атау өзгертетін, alias name бұл жарлық атауы және Linux командасы. Қарапайым мысал:

alias add = /USR / SBIN / useradd

* Терминалдық терезелерде файл мазмұнын көру үшін, бір мезгілде

экранда аз команда орындалады. Әрбір экранның соңында, сіз бір қадам алға, немесе бір жолақ артқа, немесе командалық жолаққа қайтуға болады. Файл мазмұнын қараудың оңай түрі `less` командасы жанындағы `more` және `cat`, бірақ `more` немесе `cat`-қа қарағанда `less` ыңғайлы.

`alias add` ұзақ команданы аустыру үшін қоданылатын болады. Егер Linux командасы мазмұнында кемшілік болса, онда команда жалғыз тырнақшаның ішіне алыну керек(' '). Таңдалған нышандар Linux басқа да нысандары үшін қорғалмаған кезге дейін (мысалы, `%m` лақап ат ретінде қолданылуы мүмкін, бірақ `-m` бола алмайды), атауы әртүрлі нышан жолынан тұруы мүмкін (сандар және белгілерді қоса).

Көп жағдайда сіз `alias` іске қосуды жеңілдету үшін қолданамыз. Жоғарыда көрсетілген мысалдай қолданушы басу командаларды қысқарттады. Басқаша айтқандай, таңдау мазмұнын командасының ұзақтығын қысқарту үшін. Мысалы, егер сіз қаласаңыз, сіздің `ls` командаңыз әрқашанда ұзын тізім ретінде қолданылады, сіз келесі `alias` анықтай алар-медіңіз:

```
alias ls = 'ls -l'
```

Назар аударыңыз `ls -l` арасында кеңістігі тырнақшаның қолданылуына. Сондай-ақ, `alias` атауы команда да қолданылуы мүмкін.

Егер еске түсіре алмасаңыз, `alias` сондай-ақ пайдалы болуы мүмкін, оны қалай қолданады. Мысалы, `useradd` жағдайында, сіз қайда сақтағаныңызды ұмытап қалуыңыз мүмкін. Немесе команда кезінде, сіз бірнеше нұсқаны біле тұра, нені қодану керек екенін ұмытып қаласыз. `Rm` жағдайында сіз `-i` таңдауды ұмытуыңыз мүмкін, бірақ `rm -i` әрқашанда қолдану пайдалы. Осылайша, сіз келесі `alias` анықтап аласыз.

```
Aliasrm = 'rm -i'
```

`Ls` командасы параметрлерді қажеттпейді, сондықтан сол шкала `ls -l` ашылады. Бірақ `rm` және `useradd` параметрлерді қажетеді. Параметрлер лақап аттармен қалай әрекеттеседі? Мысалы, егер біз `foo.txt` өшіргіміз келсе, `rm foo.txt` теру керекпіз. Енді бірақ `rm 'rm -i'` үшін лақап ат болады. Күтілетіндей лақап ат ауысу жұмыс іске асады. Лақап ат `rm rm -i` бастап `foo.txt` дейін жайылып нұсқаулықпен аяқталады. Осылайша `rm foo.txt rm -i foo.txt` болады.

Ұқсатық бейнесі бойынша `-m -G cit130 bearb` айналады, мынаған / `USR` / `SBIN` / `useradd -m -G cit130 bearb`. Лақап ат команда қосылардың ал-

дында, жай ғана ауысады. Rm лақап атында қосымша таңдау жасауға болады. Мысалы, егер сіз (rm -r) рекурсивті кылып істегіңіз келсе, rm -r / Foo командасы rm -i -r / Foo сияқты болу үшін айналады.

Лақап ат сондай-ақ қолданушы жиберген, ортақ қателері түзеу үшін қолданылады. Алда қауіпсіз машинамен, екі әріптік команданы елестетіп көрейік. Осылайша mvvm, rmшығады, mrgәне srмынау сияқты шығады ps. Екі әріптік комбинациялар vm, mr және psLinux командасы емес, басқаларын енгізу қате хабарламасымен командалық жолаққа келетін болады(команда анықталмады). Юрбіріне жеке лақап ат таңдауға болады:

```
Aliasrm = rm
Aliasps = ps
Aliasvm = mv
```

Команда өзін кеңістікке қоспағанда, тырнақшалар қажет емес екенін ескересіз.Алдыңғы мысалменrm идеяны біріктіре отырып, біз осы лақап аттуымыз мүмкін.

```
aliasrm = 'rm -i'
aliasmr = 'rm -i'
```

Төменде бірнеше басқа да қызықты идеялар alias ұсынылған.

alias..='cd ..'	A shortcut to move up a level
alias...='cd ../../'	A shortcut to move up two levels
alias md=mkdir	A shortcut to create a directory
alias egrep='grep -E'	Enforcing a common option in grep
alias h='history 10'	A shortcut to display a partial history
alias xt='xterm -bg black -fg white &'	A shortcut to generate a terminal window

Егер сіз лақап атта командалық жолақты анықтасаңыз қабықша сессиясында ғана болады.Егер сіз жаңа қабықша сеансын бастасаңыз, ол басқа терминалдық терезелерде болмайды(мысалы, Bash басы). Сол себептен ол лақап аттар жалпы сақтау орны.Біз Bashдара қабықшасында қалай істеу керекін қарастырамыз.

Сонымен қатар, лақап ат ұсынған қысқа командаларға, екі қысқа командалар түрлері тильда және өрнектер жақшасы арқылы табылады. Сіз қазірдің өзінде тильда (~) пайдаланушылар қатысушының үй тізімдеме үшін қалай қолданатынын көрдіңіз.Аудармашы ~ Bash пайдаланушының үй тізімдемесін ауыстырады, сондықтан, мысалы,less ~/foo.txtless

/home/foxr/foo.txt. айналады.Тильда кеңейту арқылы, сондай-ақ ~ Қатысушының аты бар басқа үй тізімдемелері көрсету үшін пайдалануға болады. Мысалы,сіз үй тізімдемесінен зарraf файлды көру үшін,less ~ зарraf / foo.txt пайдалануға болады.

Сіз команда қолданылуы тиіс элементтер тізімі бар кезде, тырнақша кеңейтілуі қолданылады. Өрнекті жақшаға, {}, тізімдегі әрбір элемент үтірлермен бөлінген тізімін қамтитын болады.Қарапайым мысал ретінде, сіздің үй тізімдемеіңізде бірнеше Ls подкаталогын жасау керек делік. foo2 екі подкаталогі foo2a және foo2b бар жерде сіздің үй тізімдемеңізге foo1, foo2 және foo3.Сіз (Foo1 үшін одноq, foo2, foo2a, foo2b, foo3 үшін) бес бөлек үшін Ls командасын босатуға болады.Сонымен қатар, сіз бір командалық LS олардың барлығын тізуге болады: Ls foo1 foo2 foo2 / foo2a foo2 / foo2b foo3.Алайда, foo2a және foo2b үшін, сіз foo2 / foo2a және foo2 / foo2b, оларды аударуға тиіс екенін атап өтеміз. Сіз оның орнына жақшаларда тізімдемелерде барлық тізіп ұзарту жақшаларды пайдалана алар едіңіз ішкі тізімдеме жақша ішкі (немесе кірістірілген) жиынтықта ұқсас жолмен көрсетілуі мүмкін командасы болады: бес бөлек команда, Ls Ls Foo1, Ls foo2, Ls foo2 / foo2a, Ls foo2 / foo2b Л.С. foo3, Bash түсіндірушісі кеңейтеді Ls {Foo1, foo2, foo2 / {foo2a, foo2b}}, foo3}.

Кеңейту тағы бір түрі файл кеңейтімі деп аталады. Егер сіз файл атауларына арасында таңбаларды қолдану кезінде орын алады. Қалауыңызша покер алмастыру, сондай-ақ, әрекет ете алады. Ең көп таралған қойылмалы мәні бойынша «ештене сәйкес келу» деген мағынаны білдіреді, ол * болып табылады. Мұндай, сіз * қолдану кезінде LS командалар жүйесі файлдарда қолдана аласыз. Мысалы, Ls .txt қарамастан, олардың аты қандай болды, ұзарту .txt аяқталады барлық файлдарды тізеді *.

Ағымдағы тізімдеме foo1.txt файлдар, foo2.txt, foo3.text және bar.doc, *.txt белгілеу foo1.txt және foo2.txt келетін болып қамтиды. Құжат TXT сәйкес келмейді, өйткені, TXT мәтіні бірдей емес, себебі foo3.text сәйкес келмейді, және ол bar.doc сәйкес емес болады. Дегенмен, Ls * командасы. *, барлық төртеуі сәйкес келеді. Сондай-ақ, файл, тізімдеме нәрселері бар болса, басқа жағынан, * LS. * Заттар бірде-бір нүктеден, өйткені оған сай емес болады. Команда Ls * барлық бес файлдарғада (төрт балдық және материал) сәйкес келеді.

Қойылмалы таңба пайдалану Linux пайдаланушылар үшін ортақ болып табылады. Мұндай білгі ? нені білдіреді, басқа да арнайы таңбалар бар «бір кез келген таңбаны.» Біз тұрақты тіркестерін қараған кезде біз, келесі тарауда қойылмалы таңба толығырақ зерттеуге болады.

Команда жолында жұмыс оңай, сондықтан команда жолының редакциялаудың түрлі әдістері, қысқа команда қуатты жиынтығымен пайдаланушымызды қамтамасыз етеді. Алайда, кейбіреулерін қоспағанда, олардың көбісі тәжірибесін қажет етеді. Олар тек Linux үйренуші студенттерге арналған интуитивті емес, өйткені негізгі Linux пайдаланушылар жиі осы әдістерді пайдалануға құлықсыз. Екінші жағынан, Linux неғұрлым тиімді болуға мүмкіндік алады, себебі ерте оқитын, оның пайдаланушылар көп білгісі келеді.

BASH ТҮЗІЛІСІНІҢ ЗЕРТТЕЛІНУІ

Біз Bash қапшығына тиісті кейбір қысқа командаларды және түзету функцияларын білген кезде Bash түзілісінің қалай жұмыс істейтіндігіне көшеміз. Алдымен, үш қосымша құралдарды енгіземіз: ауыспалы, қайтабағыттау, және анықтамаларды алу.

Bash қабықшасындағы ауыспалы мүмкіндіктер. Ауыспалы есім бұл-мекенжайдағы сақтаушы мәлімет. Ауыспалыны қолдана отырып, сіз тікелей мағынаны қолданбастан бұрын, әртүрлі командалардың мекенжайына түсе аласыз. Қарапайым мысал ретінде лицензиялардың санын сақтап отыратын, бағдарламаны қамтамасыз етуге мүмкіндік бола алатын ауыспалы бола алады. Соған қарамастан қолданушылардың бағдармаларды іске қосқанда, ол ауыспалы азая бастайды және қолданушылар бағдарламаны қамтамасыз ету жұмысын тоқтатқан кезде ауыспалы көбейеді. Егер қолданушы бағдарламалық қамтамасыз етуді іске қосқысы келсе және мағынасы 0-ге тең болғанда, олар бос лицензиялардың жоқ екендігін және оны күтулері тиістігін айтады. Бұл жағдайда ауыспалы барлық қолданушылар үшін жалпы болып қалады. Сессияда Bash осы қапшықта жұмыс істейтін ауыспалы бағдарламалық қамтамасыз етуге мүмкін болады. Bashтың ауыспалы екі типі болады. Қолданушылық ауыспалы, нақты қолданушысы бар және онымен жиі командалық жолақтар операцияларының көмегімен орындалады. Сонымен қатар, ОЖ-да орналасқан, Bash түзілісінде немесе басқа да БО құрылымында орналасқан ауыспалы қоршаулар болады. Мұндай ауыспалылар ереже бойынша барлық бағдарламалық құралдарға сәйкес келеді. Bash қабықшасы өзінің нақты ауыспалы қоршауларын анықтайды, олардың көбісі қолданушылар үшін пайдалы бола алады. Қандай ауыспалы қоршаулар орналасқандығын аңқтау үшін `env` командасын енгізесіз. Ауыспалы, анықталғандардың арасынан сіз, әрине, осы секілді дүниелерді көре аласыз: `HOSTNAME` (компьютер аты), `SHELL` (аталып отырған қабықшаның негізі, біз үшін `be / bin / bash`) `USER` (қолданушының өз аты),

HOME (қолданушының үй жайындағы тізімі) и PWD (аталып отырған жұмыс тізімі). Осы аталып отырған ауыспалылардың барлығы бас әріппен жазылатындығына баса назар аударыңыз. Бұл анықталған, сондықтан қолданушы ауыспалы қоршаулар ма, әлде басқа нәрсе (қолданушының аты мен лақап атымен осындай анықталған) екендігін бірден аңғарады.

Өзіндік жеке ауыспалыны орнату үшін, жай ғана командалық жолақта иелену операторын енгізіңіз. Формат оператора присваивания Variable = операторының форматын иелену үшін value, мұнда variable ауыспалының есімі ретінде және value мағынасы болып табылады. Bash жолақтарды сақтауының мағынасы, ол тжоллақтар мағынасы болып табылуы тиіс, мысалы адамның есімі секілді

Егер жолақтар бос ауқымды алатын болса, барлық тжоллақтарды тырнақшамен белгілеңіз(""немесе "", біз оларды кейін арастыратын боламыз.).

Мағынасы жалпы санмен көрсетілуі мүмкін; Соған қарамастан, Bash жолақтардағы кез –келген санды қарастыратын болады, егер ол сандық болу керек екендігі көрсетілмеген болса;Бұл бар мағынаны жинайтын жақшалардың ішіне алынатын болады, жастағы секілді = (29).

Ауыспалыда сақталғанның мағынасын алу үшін,сіз \$пен есімніңізді ауыспалы ретінде көрсетуіңіз керек. Мысалы, егер first = Frank и last = Zappa және сіз ауыспалыны шығарғыңыз келсе, және сіз толық атыңыз бен FULLNAME = "\$ first \$ last"оператордың иеленуіне бере аласыз. Ауыспалы FullName одан кейін мағыналы ретінде орналастырылады, бірінші ауыспалыда сақталады, пробелға жалғасатындықтан, одан кейінгі мағнасы, Lastта сақталады.Егер сіз FullName = "first last" командасын орындауға тиісті болсаңыз, онда FullName міндетті түрде "first last" деп аталады, ал мағынасы жоқтар осы ауыспалыларда сақталады. Бос кеңстіктің жағдайымен тырнақшалардың міндетті түрде керек екендігіне көп мән беріңіз. Егер сіз бос кеңестікті ойламайтын болсаңыз, сіз осы команданы орындай аласызFullName = \$ first \$ last, сонымен қатар, FULLNAME = "\$ first \$ last".

Егер мағынасы ауыспалы сан болатын болса, және сіз сол жерде бірқатар арифметикалық операцияларды орындағыңыз келсе сіз барлық операцияны \$ке ((...)) салуыңыз қажет. Рассмотрим, если, после того, как сохраним 29жасты сақтағаннан кейінгі болатын жағдайды қарастырайық, ол үлкен болып көрсетілгендіктен сіз оны NewAge = \$ age + 1 деп белгілейсіз. Сіз30 ды жаңа дас ретінде аласыз ба NewAge = \$ age + 1е? Жоқ, негізінде сіз29 + 1 деген команданы орындайтын боласыз.

Неліктен? Нақты жіктеген кезде командалар элементтерді қосып алады да, сақталынатын ауыспалы мағынасы(29) + және мағынасы бірге тең болады. Қоспасы жолақтарды бірге қосы алады. Бірақ, біз бір жасты қосқымыз келеді. Ол үшін ((...)) қолданылуы туралы ақпарат керек болады, өйткені, дұрыс құрылымы $NewAge = \$ ((age + 1))$ болып табылады. \$ тің ((...)) ұсынатынын және жастың ауыспалы кезеңінде шықпайтындығына баса назар аударыңыз. Сіздердің араларыңыздағы бұрындары бағдарламаны қолданушылар болса, өтініштің ұсынысымен таныс болуы керек, онда ауыспалының мағынасын қабылдауы, оны өзгертуі, және жаңа мағынасы үшін ауыспалы қайта қаруы кіреді. Осындай қалыппен, $NewAge = \$ ((age + 1))$ орнына, сіз сонымен қатар $age = \$ ((age + 1))$ командасын орындай аласыз.

Егер сізде екі сандық мағына болса, айтарлық, X және Y, ол кезде сіз $Z = \$ ((X + Y))$ командасын баса аласыз. Арифметикалық операцияларға Мүмкіндігі бар +, -, * (көбейту), / (бөлу) және % (қалғаны, сонымен қатар барлығы білетіндей модульдікпен).

Ауыспалылардың қойылуы мен олардың операторлық иеленуде қолданылуын сіз сонымен қатар echo командасын орындау арқылы көре аласыз. Echo командасы элементтер тізімін күтеді. Бұл тізімдемеде не болса да барлығы шығарылады. Мысалы,

echo Сәлем, қалың қалай?

Негізгі терминалдың жолағындағы келесі терезе шығарылымы «Сәлем, қалың қалай?» деп аталады. Барлық тізім бойынша орналастырудың қажет емес екендігіне баса назар аударғаныңыз жөн. Сіз сонымен қатар, келесі мағынаны енгізе аласыз, олар ауыспалыда сақталатын болады, мысалы, жаңғырықта $\$ FULLNAME$ немесе echo \$age деп аталынады. Сіз дұрыс аяқталмаған, өңделмеген текстті жүйелеп, echo Hello \$FULLNAME дегі секілді жаңа өзгерістерді енгізе аласыз., Жаңғырық нақтылықты көрсетеді, бірақ, сіздің тырнақшаға алынған элементтерге қосуыңызды талап етпейді. Сіз «немесені» қолдана аласыз, бірақ үлкен айырмашылық болуы мүмкін. Егер сіз «қолданып және атыңызды ауыспалы бірдей тырнақшаға орналастыратын болсаңыз, сіз мағынасын емес, ауысымын алатын боласыз. Сонымен, мысал келтіретін болсақ, echo "Hello \$ FullName" Сәлем, Френк Заппа(FullName Frank Заппаны сақтайды деп жобалай отырып), осы кездегі жаңғырық 'Hello \$ FullName' сонымен бірге келесі функцияны шығарады, Hello \$ FULLNAME, және ,әрине, жаңғырық "FULLNAME Hello" оңай жолы болып табылады, Сәлем FullName келесі \$функциясы оның алдында

ауыспалыны орындамас үшін қолданылады. .

Echo өтініші бойыншы соңғы ескерту. Елестетіңіз, сіз қолданушының сәлемдемесін және оның нақты уақыты мен күнін белгілеп, шығарғыңыз келді, Linux date командасын иеленеді.. Егер сіз осыны қолдануға тиісті болған болсаңыз

```
echo "Сәлем $ FullName, бүгінdate"
```

сіз сонымен қатар осыны аласыз Hello Frank Zappa, бүгінгі уақыт. Linux командасын ішіне орналастыра отырып `` (кері тырнақшалар), ол кезде команда осыecho-өтінішке дейін орындалынатын болады. Сонымен,

```
echo "Сәлем $ FullName, бүгін date` "
```

осы секілді команда орындалынатын болады Hello Frank Zappa, бүгін бейсенбі,9 ақпан сағат: 17:54:03 EST 2012 (бұл команда сол уақыт пен сағатқа байланысты дәл енгізіоген болатын). Команда « қалып, мүмкін келесі форматта болуы әбден мүмкін \$ (команда Алдыңғы команда жаңғырығы болу үшін орындалады Предыдущая команда эхо может вместо того, чтобы быть

```
"Сәлем $ FullName, бүгін $ (date)"
```

Қолданушыға өзінің жасы мен атын өзгерту мақсатында, ауыстыру командасын осы мақсатта қолдану керек деген қағида жоқ. Қолданушылар ауыспалыларды анықтауы мүмкін, ол олар жазып жатқан бағдарламалық қамтамасыз етуде қолданылады.Көбінесе, ауыспалылар осы бағдарламалық қамтамасыз етумен анықталынады және бағдарламалық қамтамасыз етумен қолданылады. Linuxтің бірнеше ауыспалы орналастырылған орталары бар.. Мы видели некоторые из них раньше, когда мы говорили о команде Біз ол туралы env командасы туралы айтқан кезде көрген болатынбыз. Міне, басқа да ескертулер

- DISPLAY-ия Linuxтің стандарттық шығарылымындағы аты(негіз бойынша,: 0.0).
- HISTSIZE-тарих тізімінде сақталынуы мүмкін командалардың тізімі.
- PATH-қолданушы команданы орындаған кездегі каталогтар тізімі, олар күн сайын тексерістен өтетін, ;Бұл қолданушыға бірқатар бағдарламаларды орындауға және кейбір файлдарды табуға көмектесіп, жол

көрсетеді;

- PS1-қолданушының сұрағын анықтайды, ол тарих тізімінде сақталатындықтан өзіне кейбір символдарды қосуы мүмкін, онда қолданушының ат-жөні немесе аталған катлог, немесе команда номері болуы мүмкін.(таблица 9.1)

- PS2, PS3, PS4, олар басқа да түрлі бағдарламаларда қолданылатын мүмкіндіктерді анықтаулары мүмкін;

Ауыспалылардың, санлатындардың ішінде ең жоғарысы, яғни, сіз тек PATHды ғана өзгерте аласыз. Ауыспалы PATH каталогтар тізімін сақтайды. Әр икезде, сіз командаларды енгізген кезде, немесе Linux файлын атын енгізгенде, егерде элемент аталынып отырған жұмыс үстелінде табылмайтын болса, онда түзетушіPATHда көрсетілгендей Linuxтің әрбір директориясын тексереді.Бірақ, ол аталынып отырған жұмыс үстелінде немесе табылмай қалған жағдайда PATHда кез-келген каталогта қателіктер туралы хабарлама келмейді. .

Кесте 9.1 Ауыспалылар үшін символдар PS1 қолданушының мүмкіншіліктері үшін

Таңбасы	Мағынасы
\ d	Аталған уақыты
\ d {format}	Аталған уақыты, осы форматта қолдана отырып, күтілетін команданың STRFTIME форматымен сәкес болуы керек
\ t	Аталған уақыт әскери форматта болады
\ @	Аталған уақыт 12 сағаттық режимде (AM / PM)
\ H, \ h	Компьютердің толық аталынатын аты
\ s	Қабықшаның аты
\ u	Қолданушының аты
\ w	Аталған жұмыс каталогы
\ !	Аталған команда номері (тізімде көрсетілгендей)
\$	Қолданушы негізі спецификатор болып табылады (тез қимылдайтын)
@	Бөлуші, мысалы, \ d @ \ u \$
\ \$?	Соңғы команданың статусы
\ e [... \ e [m	Жолақтардың түсін өзгерту, сонда ... бұл түстің түрі бойынша коды #; ##, бірінші # немесе 0 (светлый) немесе 1 (сұрғылт), ал екіншісі ## түс болып табылады (30қара үшін, 31 қызыл үшін, 32 жасыл үшін, 33 қоңыр түс үшін, 34 көк түс үшін, 35 күлгін, 36 көгілдір түс үшін)

Bash қабықшасында, ауыспалы PATH алғашқыда жүйелі администратормен орындалады, типтік сақтау: /USR /kerberos /bin: /USR /local /bin: /USR /bin: /bin: /USR /X11R6 /bin. ":" Әр жолдың бөлінуі үшін қолданылады. Сіз осы жолдардың көбісін сіз мойындау керексіз (мысалы, /USR /local /BIN, /USR /bin, /bin). Kerberos bin каталогында орналасқан және осылар секілді (например, FTP, Telnet, Rlogin) жалпы желілік бағдарламада орналасатын желілік аутентификация мен бағдарламада қолданылады. X11R6 катлогтарды X11 (оконные) файлында сақтайды.

Егер сіз ауыспалы PATH ды қосқыңыз келсе, сіз қайта ұсыну өтінішін келесі жолмен берер едіңіз:

`PATN = $ PATH: newdirectory немесе`

Сіз бұл жерде PATHды түзу етіп орналастырасыз, себебі дәл қазіргі уақытта ауыспалы PATH, оған келесі жаңа каталог ереді. Сонымен, `PATN = $ PATH: /home /foxg` болған кезде `add /Home /үй` жайындағы каталог `foxg` сіздің жолыңызда болады. Сіз оны өзіңіздің жеке үй жайындағы директорияңыз ете аласыз, сол себепті, сіздің Linux файлдық жүйесінде жұмыс жасайтындығыңызға қарамай сіздің үй жайындағы каталогыңыз басқа да PATHдағы каталогтармен тестілеуден өтетін болады.. Сіз, әрине, өзіңіздің жеке қолданушылық есіміңізді `foxg`дағы жылдар үшін өзгертер едіңіз.

PS1та ауыспалы сақталады, осы кезде сіздің сұрағыңыз қалай көрінетін болады. PS1өзгерте отырып, сіз өзіңіздің мүмкіншілігіңізге қарап ішкі сипатыңызды өзгерте аласыз.Егер сіз өзіңіздің не істеп жатқандығыңызды білмесеңіз, оны PS1 жайына қалдырғаныңыз жөн. Сонымен қатар, ауыспалы қоршаған жерлерді де өзгертуге керек болады.

Bash түзілісі ереже бойынша пернетақтадан кірме алады және терминал терезесінен шығарады. Соған карамастан басқа да жағдайлар болып тұрады,мысалы, сіз не шығарғыңыз келеді, не кірмеге басқыңыз келтін жағдлайлар. Мысалы,сіз командалық жолақта емес, бағдарламаға кіру рәсімдеріңізді өзгеруіңіз мүмкін,Немесе сіз дисплейдің орнына файлға шығарылымыңызды жібере аласыз

Үшінші мүкіншіліктің мағынасы: сізбір командадан шығарылымды алып және оны басқа командалардың есебіне қолданатын боласыз.Бұл формалардың барлығы қайта бағыттау болып табылады..

Қайта бағыттау келесі 5 таңбалардың біреуінің қолданылуымен жүзеге

асады.:

- <кіретін файлдан кірме қайта бағыттау командасын орындау
- >шығатын файлдан шығару бастырмасын басып, файл болатын болса, жазып алу керек;
- >>келесі файлға қосылатын жағдайда немесе оны құрастырған кезде шығарылатын файлды қайта бағыттау
- <<пернетақтаға енгізуді қайта бағыттау, онда кірме арнайы сөздерменен аяқталады\\: «құпия сөз», «көрсетілгеннен кейін»<<
- | бір команданың шығарылуын қайта бағыттау

Төрт үлгіден кейін файлды қайта бағыттайды, себеі олардың стандартты кірмесі мен шығарылымын қайта анықтап алу қажет.Мысалы, стандартты шығарылым мысық үшін экран болып табылады.Cat fool.txt тере отыра, бұл жердегі шығарылым монитормен fool.txt осылай қамтылады.

Сіз оның орнына пернетақтадан кірісін қабылдау кезінде, файлдан кірісін алатын бағдарламаны жасауға қажет болғанда << пайдаланыңыз. Егер сіз << пайдаланатын болсаңыз, бағдарламаның элементтері енгізуге мүмкіндік беру үшін кідіреді. Егер сізден үлес түсетін болса, онда қалай бағдарламасы оны біледі? Егер сізде енгізу үзілістер болуы мүмкін болса, себебі бұл жай ғана, <Enter> пернесін басу емес. Оның орнына, << кейіпкерлері кейін, сіз кілт сөзді қойыңыз. Сол кірісті аяқтайтын негізгі сөз болып табылады. Cat файлды құру үшін пайдалану << бірыңғай үлгісі. cat, әдетте, cat, файл атауынан (лар) үшін тағайындау. Егер сіз cat << кілт сөзді орындаңыз және егер сіз кілт сөзді енгізіңіз дейін алайда, содан кейін кіріс, пернетақтадан келеді. Мынадай ескертпе:

cat << done

Бұл> кілтін пайда әкеледі. Командалық жолда теруді бастаңыз. Сіз <Enter> пернесін бассып, оның ішінде, қалаулы ештеңе енгізуге алмайсыз . cat бағдарламасы, сіз енгізу жасай бастағаннан дейін де жазбаларды қабылдауға жалғастыруда. Бұл ретте, сіз жай ғана қандай хабар тересіз, ол экранда пайда болады, яғни, стандартты шығысына енгізу мазмұнын көрсету үшін жасалады, cat оның кіреберіс барлық алды, және ол қазір қызмет көрсетеді. Бұл пайдалы емес. Алайда, сіз шығару бағыттауда боласыз. Сіз істеу керек едіңіз:


```
cat << done> shoppinglist.txt
```

және сіз сөздер «Дайын» аяқталатын, сауда тізіміне енесіз, содан кейін сіз жинаған, барлық shoppinglist.txt файлында сақталады. Осылайша, cat жылғы, және қайта бағыттауды пайдаланып, сіз өз TextFiles жасай аласыз. > Белгілері терминал терезесінде көрсетіледі мынаны көріңіз:

```
cat << done> shoppinglist.txt
```

```
> банан  
> алма  
> нан  
> сүт  
> аспирин  
> done
```

Сіз банан мәтіні бар файлды табу керексіз, shoppinglist.txt, алма, нан, сүт, аспирин, жеке жолда әрбір, бірақ сөз done.

Файлға деректерді жіберу үшін ең көп таралған пайдалану айтамыз. Файлды сақтағыңыз келесе, мысалы, келесі әрекеттердің бірін орындауға болады:

```
ls -l> directory_listing.txt Ls -l >> directory_listing.txt
```

Файл бар болса, соңғы жағдайда, командалық файл жасалынады және оны қайта жазбайды. Соңғы нысанды қайта бағыттау құбыр деп аталады. Құбыр басқа бағдарламаға кіру үшін бір бағдарламаның шығысын қайта бағыттайды. Қарапайым мысал ретінде, ағымдағы каталогтың мазмұнын көру үшін Ls пайдалануға болады.

Алайда, тым көп файлдар көре аласыз, олардың соншалықты көпшілігі тым тез экранды өшіріп қалдырды. Егер сіз Ls экрандар арасында кідірту үшін алғыңыз келе ме. Сіз көп немесе аз таңдау арқылы жасауға болады. Сөйтіп, сіз мыналар сияқты арналарды пайлана аласыз:

```
ls | more  
немесе ls | less.
```

Команда Ls бірінші орындайды, бірақ оның орнына шығысын визуализацияланған, бағдарлама, көп (немесе аз) шығаруды бағыттайды. Сонымен қатар, бұл проблема әлдеқайда айналма жолын шешуге бола-

тын болған. Теру арқылы:

```
ls> directory_listing.txt
```

содан кейін енгізу

```
less directory_listing.txt
```

Құбыр екі бөлек пәрмендерді жасауға сақталады, және сіз шынымен саған қажеті жоқ файлды құру сізді арылтады.

Linux құбырды жиі пайдалану ретінде талап етеді. Бұл ықтимал ұзақ және өте күрделі нұсқауларға әкелуі мүмкін. Келесі амалдарды қарастырайық:

```
ls -l | -fr sort | less
```

Бұл жағдайда, сіз кері алфавит бойынша тізімдегі элементтерді сұрыптап, ағымдағы тізімдеменің ұзын тізімін алуға, сондай-ақ тәуелсіз (ол жоғарғы және төменгі жағдайда арасындағы түрлі болуы мүмкін емес болады), содан кейін кем пайдаланып нәтижесін көрсетеіз.

Біз назарға Bash қол жетімді көмек ерекшеліктерін ескере отырып секциясын аяқтаймыз. Шын мәнінде, бұл функциялар кез келген Linux қабығынан табуға болады. Осы құжаттың сипатталған түрлі Linux пәрмендерін пайдалану жолын ашып көмектеседі. Көмек ең көп таралған оқыту формасы үшін беттер (ер) басқарылуы болып табылады. Қолмен беттерді көру үшін, табудың екі бірдей адамның немесе тұлғаның заттай, командасының адам нысанын қабылдайды ер пәрменін пайдаланыңыз. Көпшілік беттер үшін, әрбір Linux командалары бар. Адам беттер пәрменін пайдалану жолын түсіндіреді көмектеседі түрлі ақпарат түрлерін береді. Оларға мыналар жатады:

Оны пайдаланудың қысқаша сипаттамасы (синтаксис)

Сипаттау командалары

қол жетімді опциялар нұсқасы (Опциялар)

бағдарламасының авторлары

Пәрмені және басқа да қатысты пәрмендері бар файлдар, немесе осы сияқты

Кейбір мысалдар қарастырайық. Біріншіден, біз RM нұсқауларды қа-

раңыз. Біз ер RM қонырау кезде, бірінші суретте 9.3 көрсетілген экранын қараңыз.

Беттің бірінші экран RM кісінің есімі, оны пайдалану қысқаша шолу үшін нұсқауларды және алғашқы екі опциялардың сипаттамасын тізімдейді. «:» Түбі терезе көрсетеді және кейін меңзер. Адамның бетіне трафикті жетек үшін, біз, алға жылжу үшін экранда дисплейді ғарыш жолағын пайдаланамыз.

```
RM(1)                                User Commands                                RM(1)
NAME
rm - remove files or directories

SYNOPSIS
rm [OPTION]... FILE...

DESCRIPTION
This manual page documents the GNU version of rm. rm removes each
specified file. By default, it does not remove directories.

If a file is unwritable, the standard input is a tty, and the -f or
--force option is not given, rm prompts the user for whether to remove
the file. If the response is not affirmative, the file is skipped.

OPTIONS
Remove (unlink) the FILE(s).

-f, --force
    ignore nonexistent files, never prompt

-i, --interactive

:
```

Сурет 9.3 бірінші rm man бетінің экраны

«Q» әріпі жұмысты аяқтайды. Нұсқалардың және ары қарай келесілер екі экранда (тек бөлігі суретте 9.3 көрсетілген) .Көрініс 9.4 суретте көрсетілгендей, қалғаны қалай көрінеді.

RM бетті қараған кезде, біз нұсқалар **-f**, **-i**, **-r**, немесе **-R** және **-v** екенін көруге болады. Біз сондай-ақ **--interactive** пайдалану орнына **-i**, **-f** немесе оның орнына **--force** мүмкін. Т.б. бар **--help** және **--version** үшін опциялар бар және т.б.

RM командасы қарапайым. man Find командасымен неғұрлым күрделі. Шын мәнінде,man find беті 50 экранды болады.

Бірі елеулі айырмашылықтарды беттерінде басшылығының екі "командалары бұл олжа қамтиды тізімі, мысал үшін, суреттейді, қалай пайда-

лануға болады команда. Дегенмен мысалдары болып табылады жалпы адам беттерінде, сіз болған кезде, оларды командасы өте қиын. Тағы бір вариация беттерінде басшылығының табуға болады көмегімен анықтамалық бетіне бекітуге арналған. Ал `tt` табу тек бір пайдалануға синопсис, бекіту бар төрт. Олар мыналар:

бекіту [-lhV]

бекіту -a [-ffnrsvw] [-tvfstype] [-ooptlist] монтаждау [-fnrsvw] [-o опциялар [, ...]] құрылғысы | реж

бекіту [-fnrsvw] [-tvfstype] [-o опциялар] құрылғысы реж

Осы төрт жолды көрсетеді, - деп бекітілуі мүмкін төрт түрлі типтері параметрлерін. Біріншіден, бекіту бара алады бір немесе одан да көп `l`, `c`, немесе `V` опция ретінде. Екіншіден, бекіту -a жүруі мүмкін опциялардың тізімімен, үлгідегі виртуалды файлдық опция ретінде. Екіншіден, бекіту -a жүруі мүмкін опциялардың тізімімен, үлгідегі виртуалды файлдық жүйесі, сондай-ақ кез келген қосымша опциялар үшін осы команда. Үшіншіден, бекіту жүруі мүмкін опциялардың тізімімен, қосымша опциялар, сондай-ақ құрылғы немесе каталог (символы | білдіреді құрылғы немесе каталог тиіс нұсқауларды ұсынылған). Ақырында, бекіту жүруі мүмкін опциялардың тізімімен, түрі виртуалды файлдық жүйесі, опциялар тізімі, құрылғылар, және каталог. Көптеген нұсқаулар бар бірнеше жазба астында синопсис. Басқа, бірақ неғұрлым қарапайым, мысал табуға болады `useradd` команда.

Екінші нысаны көмекті қолжетімді командасы. Команда синтаксисі анықтама көмек көрсету болып табылады. Дегенмен, біз өте аз команда `Linux` бар анықтамалық беттер, бұл пайдалану ең жақсы адам.

Түпкілікті көмек түрі міндетті емес көмектеседі, сонымен қатар пайдалануға болады команда. Оның орнына, ол сізге көмектесе алады тобын табу қажет. Проблемалардың бірі, `Linux` деп еске алады барлық-команда бойынша". Егер сіз білетін аты қолбасшылығы, онда бет адам көмектесе білуге, оны қалай пайдалану керектігін. Бірақ бұл, сіз білесіз, сіз қол жеткізу үшін, бірақ білмеймін, команда атауы? Бұл жерде айтпақшы кіреді. Синтаксис командасының жолы болып табылады, онда байланысты жол болып табылады жолмен, сіз іздеген. "АЙТПАҚШЫ командасы шығарады тізімі барлық командалар бар тиісті жолына, табылған сипаттамада команда қатысты.

Мысалы, командасы бар, ол туралы хабарлайды ағымдағы пайдалану виртуалды жады.

Сіз еске команда атауы. Сіз енгізген

бойынша "виртуалды жады"

По поводу командасы қайтарады мынадай:

mgemar (2) - қайта көрсету виртуалды мекен-жайы жад

vfork (3P) - жаңа процесс; алмасуға, виртуалды жад vmstat (8) - Есеп статистика виртуалды жад

Бинго, соңғы пункті болып табылады, онда, біз іздеген! Дегенмен тырнақша виртуалды жад қажет емес, оларға, айтпақшы болады возвращать не сәйкес келеді, не немесе виртуалды жад. Мұның еді тізімі ұзақ зерттеу. Дегенмен, тырнақшаға, айтпақшы қайтаруға болады тек дәл сәйкес келген. Егер біз табу үшін команда, кімнің сипаттау болды екеуі де сөздер және виртуалды жад, бірақ міндетті түрде осындай тәртіппен, біз қолданғысы келеді бағамдар.

Ретінде басқа мысал, сіз табу үшін команданы алып тастау үшін файл. Сіз типті файл жөнінде жою, ол қайтарады ұзын командалар тізімі. Сондықтан тырысасыз бойынша "файлды жою". Бұл жолы бойынша қайтару:

жоқ файлды жою үшін: ештеңе ыңғайлы

көрсете отырып, бұл жөнінде таба алмадық матч "файлды жою". Пайдалану АЙТПАҚШЫ болуы мүмкін қиын, бірақ бұл болып табылады бағалы ресурс.

Енді біз көрген әртүрлі функциялар қолжетімді Bash қабығының, біз қорытындыға келіп отырмыз бұл бөлім қысқаша шолу ретінде жұмыс істейді, аудармашы. Сіздің енгізген команданы командалық жолда. Не болып жатыр қазір? Қолдануда келесі қадамдар:

- интерпретаторлар оқиды кіру файл.
- Аудармашы бүйректегі іске қосу сөздер мен операторлар
- Операторлар болуы мүмкін рәміздері сияқты, <, >, |, ~, *
- Егер команда қандай да бір бағамдар, бұл бағамдар өңделеді.
- Нұсқаулықты өңделеді үшін қандай да бір белгілі псевдонимов, және егер қандай да бір лақап табылса, кеңейту, бүркеншік есімі (ауыстыру).
- Әр түрлі сөздер мен операторлар енді сынған жекелеген команда (мысалы, егер көлемдегі операторлары бар, онда командалар бөлінуі осы операторлардың).

- Shell ыдырау орын алады, егер қандай да бір шақырылады үшін
- Кеңейту Brace
- Тильды
- Аудио беріледі
- Егер қандай да бір команда пайда болады, не немесе ` \$ (), командасын орындау.
- Орындау кез келген деп аталады үшін арифметикалық операциялар.
- Орындауға кез келген көлемдегі (қоса алғанда құбырлар).
- Орындауға файл атының кеңеймесі бар сәйкестену рәміздер қою .
- Команда орындалады және аяқталғаннан кейін, мәртебесі шыққан командалар (қажет болған жағдайда) көрсетіледі терминал терезесінде (егер қайта шығару емес шақырылып).

Бұл естіледі ретінде көп, бірақ бұл өте тез жүреді. Енді сіз неге көмегімен ком - MAND желісі Linux сізге үлкен икемділігі мен қуатын пайдалана отырып, графикалық интерфейс.

жекешелендіру сіздің Баш Shell

Көрсетілгендей екі Bash Редакциялау Shell Ерекшеліктері және зерттеу Bash интерпретатора, сіз енгізуге командалар псевдонимді анықтау және ауыспалы пәрмен жолынан. Өкінішке орай, кез келген осындай лақап немесе айнаымалылар белгілі болады ғана шегінде осы нақты сессиясының қабығының. Сіз не дейсіз берілген бүркеншік аты немесе айнаымалы, белгілі әрбір сессиясының қабығының? Бірнеше файлдарды, пайдаланушыға қол жетімді, олар жүктеледі әрбір іске жаңа сеанс қабығының. Егер сіз қойыңыз, өзінің бүркеншік аты немесе анықтау айнаымалылардың бірі-бұл файлдар, онда бүркеншік аты немесе айнаымалы белгілі болады, әрбір сессия қабығының. Міне, біз қараймыз және осы файл. Бұрын біз қарастырамыз, бұл файлдар, қарастырайық, бұл кезде басталады сеанс қабығының. Іске қосу кезінде қабығы, ол арқылы өтеді процесс баптандыру. Интерпретаторлар Bash орындайды бірнеше сценарийлер бар қабығы. Оларға мыналар жатады, тәртібі, / және т. б. / профиль, / және т. б. / Bashrc, .bash_profile, .bash_login, .profile және .bashrc. Алғашқы екі сценарий белгіленеді жүйелік әкімші. Олар әдепкіде, барлық пайдаланушылар үшін, мысалы, құру арқылы псевдонимов, олар барлық пайдаланушылар болуы тиіс және ауыспалы жолында барлық пайдаланушылар бастау Сценарий профиль орындалады бірлесіп әрқашан кез келген түрі старт қабығының болса, сценарий Bashrc орындалады кезде ғана қабығы Bash іске

қосу. Соңғы төрт сценарий айқындалады пайдаланушы өзінің үй каталогында.

`.bash_profile` Сценарий, әдетте, үшін пайдаланылады не жүйесінде белгілейді сценарийде, сондай-ақ жекешелендіру элементтері (лақап, ауыспалы) үшін пайдаланушы. Файл `.bash_login` ғана орындалады, егер қолданушы алдымен жүйеге кіру (қашықтықтан) қарағанда, жаңа қабық. `.profile` Скрипт орындалады әрқашан кез келген түрі, қабығы ашылған ғана емес, Bash.

`.bashrc` скрипт орындалады, егер жаңа Bash қабықшасы ашылғаннан кейін жаңа терезеде. Қарапайымдылық үшін, бұл, бәлкім, жай ғана жақсы өзгертуге скрипт `.bash_profile`, ол кезде пайдаланылады, әрбір іске қосу кез келген Bash.

Егер осыны талдайтын болсақ бұл файлдар, сіз көптеген жазба. Бар лақап және айнымалылар. Бірақ қабығының операцияларды сценарий. Өйткені қарастыратын боламыз қабығы `script` - ИНГ 14-тарауда, біз ескермеу мазмұны осында. `.bash_profile` Сценарий мынадай екі операциялары:

көзі ~. / көзі Bashrc ~. / `bash_login`

Нұсқаулық көзі пайдаланылады орындау үшін сценарий. ~ Көрсетеді, бұл сценарий орналасқан сіздің үй каталогына. Жазу./ білдіреді орындау "бұл сценарий". Осылайша, бұл екі нұсқаулықтар, деректер орындалады екі сценарий `.bashrc` және `.bash_login`. Егер бұл екі команда бар сіздің `.bash_profile` сценарийлер, онда, шын мәнінде, сіз редакциялауға кез келген осы үш файлдар, және олар орындалуға іске қосу кезінде жаңа қабығының Bash.

Көрейік, сіз орналастыру бірі осы файл дейміз `.bashrc` файлы. Бүгінде мынадай болады екі топ:

экспорт PS1 = "..." // барлық, сіз тез жүріп жатыр ішіндегі бағамдар
экспорт PATH = "\$ PATH: ~/ : ~/ бен: ~/ скриптер: / `USR` / `SBIN`"

Деген сөз "экспорт" дегенді білдіреді айнымалы анықталады мәні белгілі болуы барлық ортада қабықтары емес, тек ішінде осы сценарий. Көшпелі PS1, белгілі төгіледі " өз талғамыңыз. Мысалы, сіз пайдалана аласыз \ `D` көрсету үшін, күнді немесе \ `W` көрсету үшін ағымдағы жұмыс каталогы. Операторы меншіктеу қабылдайды соңғы мәні айнымалы `PATH` айқындайтын, жүйелік әкімші файл / т. б. / профиль қосады қосымша каталогтар оған. Көру үшін, егер айнымалы `PATH` сақтайды,

сіз енгізілсін жаңғырығы.

\$ PATH. Егер сіз анықтау үшін одан әрі айнымалылар, сіз оны жасай аласыз, бірақ сіз көз жеткізу үшін барлық айнымалылары бар экспорт сөзді, алдыңғы олардың болатындай, белгілі болғандай сіздің ортаға Bash қабығының.

Сіз сондай-ақ таба аласыз кейбір лақап қазірдің өзінде анықталды. Сіз редакциялау немесе қосу осы псевдонимде. Мысалы, көрсетуге болады лақап, түзетулер жаңылыс жазудың, немесе сіз лақап көмектесетін санын азайту пернелерді басулардың сипатталғандай, бұрын осы тарауда. Бір соңғы пікір. Егер сіз бір кездері өзгертуге бір немесе бірнеше осы файлдар, және сіз кейінірек тауып, бұл сіз жасаған, қателер немесе ұнайды жоқ болса, сіз аласыз көшіріп, жаңа нұсқасы әрбір осы файл. Оларды табуға болады каталогы / және т. б. / SKEL.

Мәтіндік редакторлар

Мәтіндік редактор vi: Мәтіндік редактор vi өзіндік интерфейстен тұрады. Көптеген және түрлі редакторлеу командалары, мәтінді форматтау, файлдармен операциялар т.с.с түрлі пернелер комбинацияларымен байланысты. Одан басқа vi үлкен кең мүмкіндіктерге ие. Ол UNIX тың барлық нұсқаларында болады және vi UNIX ОЖ бірнеше басқа да бағдарламаларымен байланысты. Егер үндемеу жолымен редакторлеу қажет болған жағдайда vi осы бағдарламалардың ішінен шақырыла береді. Жалпы айтқанда vi редакторымен жұмыс сиқырлы іс қимылдарды еске түсіретін сияқты — алдын ала күрделі түсініксіз дуаларды жаттап алу қажет, егер жаттап алсаң барлығын да оңай жасауға болады! Vi редакторы экранды редактор болып саналады, яғни барлық өзгерістер монитордың экранында көрініп отырады. Барлық командалар бір немесе бірнеше пернелерді басу арқылы беріледі де редактор сол командаларды орындайды, қолданушы енгізілген команданы орындату үшін Enter пернесін басудың қажеті болмайды. Бірнеше командалар қосымша редакторлеуге болатын мәтінді талап етеді.(мысалы: іздеу үлгісінің тапсырмасы). Осындай командаларды орындау үшін мәтінді енгізіп болғаннан кейін қолданушы Enter пернені басу қажет. Vi редакторын жүктеу төмендегідей: # vi [ключи] [имя_файла1...] Редактордан шығу пернені басу арқылы орындалады (Қос нүктені басқаннан кейін әріптер экранның төменгі қатарында көрінеді): Егер файл өзгертілмеген болса:: Q Enter; Егер файл өзгертілген болса, онда файлдағы өзгерісті сақтау үшін:: W Q Enter; Егер файл өзгертілген болса, бірақ қолданушы өзгерісті сақтауды қажет етпесе:: Q ! Enter. Жалпы жағдайда vi редакторына бірнеше файлдарды жүк-

теуге болады. Үндемеу жолымен бірінші файлмен ашылған редакторда сол файл аргументтер тізімінде бірінші болады. Файл тізімі болмауы да мүмкін. Бұл жағдайда дискіге жазу кезінде атын беру қажеттілігі туындайтын аты берілмеген буфер құрылады. Vi шақыру кезінде келесі кілттерді қолдануға болады: +N — файлдың N-ші қатарына көрсеткішті орнату; +/образец — файл жүктелгеннен кейін үлгі бойынша іздеу жүргізу және курсорды берілген үлгіден тұратын бірінші қатарға орнату; Одан басқа бірнеше команда: # view имя_файла файлды көру режимінде жүктейді. Бұл жағдайда редакторлеу операциясы рұқсат етілмейді. Emacs редакторы: UNIX те дамыған мәтіндік редакторлардың ішінде emacs редакторы. Редактордың құрылған кездері UNIX PDP-10 компьютерлерінде де жұмыс істеп тұрған уақыттан басталады. Emacs құрған Ричард Столлмен. Vi редакторы сияқты, emacs бағдарламалаушылар үшін жасалған болатын. Уақыт өте келе оған қосымша мүмкіндіктер ендіріле бастады. Мысалы: электронды поштаны дайындап жіберу. Emacs-тің түрлі операциялық жүйеге арналған нұсқалары бар. Солардың ішінде көп тараған нұсқасы GNU emacs (версия 20), біз осы нұсқасына сүйенеміз. GNU emacs жалпы мінездемесі келесі мүмкіндіктерге ие болады: кою, мәтінді өшіру, оны ауыстыру т.с.с түрлі редакторлеу операцияларын орындау. Символдармен және мәтіннің үлкен құрылымыдық бірліктерімен операциялар жасау: сөздермен, сөйлемдермен, тараулармен; Бірнеше файлдармен бір уақытта жұмыс істеу мүмкіндігі; Бағдарламаны дайындауға арналған, арнайы режимдердің бар болуы (бастапқы мәтінді теру, компиляция және кері шегеру), электронды поштамен жұмыс т.с.с Арнайы символдар жиынтығын қолдану мүмкіндігі. Оның ішінде ұлттық алфавиттер және т.б. Редакторды жүктеу төмендегі команда бойынша орындалады: # emacs [ключи] [имя_файла\ мәтіндік және графикалық интерфейс. Интерфейс и стиль работы с emacs жұмыс істеу стилі мен интерфейсі ол қандай режимде шақырылғанына байланысты. Мәтіндік режимде командалар кезегі мен тышқан қолданылады. Командалық кезекте Esc және Meta пернелері қолданылады («метапернелер»). Бірнеше терминалдарда Meta пернесі шындығында да болады, Бірақта оны алмастыру ретінде көбінесе Alt, Esc пернесі немесе екі перненің кезекпен орындалуы — Esc және тағы бір пернелері қолданылады. Бұл жағдайда Meta+Q жазбасын Esc Q деп түсіну қажет. emacs командалары функция түрінде Lisp тілінде таратылған. Бұндай әрбір функция аты болады, ал шындығында командалар кезегі осы функцияларды жылдам шақыру үшін қолданылады. Функцияның аты «минус» таңбасымен бөлінген бірнеше ағылшын сөзінен тұрады. Сонымен, мысалы, emacs та қолдануға болатын шақыру функция аты (кириллица әріптерін қосқан-

да): standard-display-european Көптеген командалар пернені басу арқылы орындала бермейді, редактордың минбуферіндегі команда аргументтері немесе команда атын теру арнайы Meta+X пернелер комбинацияларын енгізуді талап етеді. Минибурде көрсетілетін және терілетін мәтін немесе команда аттарын біз командалар кезегін жазу кезінде кіурсивпен ерекшелеп аламыз. Мысалы: Meta+%образецEnterмекcmEnter Бұл жерде Meta и % пернесін басқаннан кейін минибуферда шақыру пайда болады. Оған мәтін үлгісін теру керек. Enter пернесін басқаннан кейін тағы басқа шақыру пайда болады және келесі мәтінді минибуферде сол сияқты теріп болғаннан кейін Enter пернесін басып аяқтау керек.

Жалғыз пернесін басу арқылы команда бұл режимде пернесі шығу, шығып осы режим оралтуға командалық режимі. Деген режиміне Ұқсас режиміне қою; Дегенмен, сіз мекен-жайын басу пернелер, рәміздер қайта жазу рәміздер қазірдің өзінде бар. Ретінде және кірістіру, шығу пернесі қайтарады сізді командалық режимі. Үшін білуге, қай режимде сіз қараңыз төменгі бөлігінде редакторы, сіз не көресіз "--INSERT--" қазақстан "--REPLACE--" немесе ештеңе (командалық режим). Бар көптеген командаларды енгізу үшін не кірістіру немесе ауыстыру режимі. Олар-кестеде келтірілген 9.2. Себебі әр түрлі команда, бұл олардың әрқайсысы болады орналастыруға меңзерді басқа орын. Мысалы, 'мен' өзін меңзерді кірістіру жүгіргі, 'о' ағымдағы бос жолына жолында курсорды төмен және ағымдағы курсор басында бос жолдар, және 'О' ағымдағы бос жолды жол алдында курсор. Үшін қосу режимін басқару үшін, кірістіру немесе ауыстыру, пайдалану,

бір әріптерден 9.2-кестеде, және қайтып командалық режиміне, пернесін басып, шыққан.

Басқа жайғастыру курсорды ауыстыру кезінде режимі кірістіру, бар көптеген басқа да тәсілдері үшін курсорды жылжыту. Олар кестеде 9.3. Назар аударыңыз, басу пернелер курсорды жылжыту үшін бір позицияға (төмен, жоғары, солға, оңға) жасалуы тиіс арқылы J, K, H, л басулардың пернелер емес, пернетақтамен қозғалту үшін көрсеткі пернелерін қолданыңыз, олар мүмкін немесе мүмкін емес бейнеленген VI / Vim арқылы G немесе #G (#нөмірі) ауыстырады сізді жаңа нүктесін файлда тез; Дегенмен, #Gтребует, сіз білесіз бе жолының нөмірі. Іздестіру командасы болуы мүмкін пайдалы.

9.2-Кесте Командасының режимі

Басатын пернелер	Сипаттамасы
i	Курсордың сол жағынан insert басыңыз
a	Курсордың оң жағынан insert басыңыз
o	Ағымдағы жолдан кейін кеңістік пернесін басып insert басыңыз
O	Ағымдағы жолдан бұрын кеңістік пернесін басып insert басыңыз
I	Ағымдағы жолдың басында insert басыңыз
A	Ағымдағы жолдың аяғында insert басыңыз
r	Енгізілген бір символды екінші символға ауыстыру кезінде енгізіледі
R	Ауыстыру пернесін басып және символдарды ауыстыруды (қайтадан жазуы) жалғастырыңыз (escape пернесі басулы тұрғанда)

9.3-кесте Курсорды жылжыту және іздеу командалары

j,k,h,l	(нұскалар істемеген уақытта) Курсорды төмен/жоғары/солға/оңға қозғау
w,b	Бір сөзбен курсорды алдыға/артқа жылжыту
G	Файлдың соңғы жолағын жылжыту
IG	Файлдың алдыңғы жолағын жылжыту
*G	* жолына жылжыту(ондағы * 5G немесе 50G сияқты сандар)
H,L	Экранның басына/аяғына жылжыту
0,\$	Жолдағы бірінші соңғы символға жылжу
/text<enter>	Мәтіннің бас жағын іздеу (мәтін қалыпты өрнектерді қамту мүмкін)
/<enter>	Соңғы іздегенді қайтадан кездесуін іздеу
?text<enter>	Мәтіннің аяқ жағын іздеу
?<enter>	Соңғы іздегенді қайтадан кездесуін іздеу
<pg up> or cntrl+f, <pg down> or ctrl+b	Экранда файлды жоғары немесе төмен жылжыту

Сіз атап өту мәтін, содан кейін көшу бөлінген мәтін. Бұл пайдалы редакциялау үшін үлкен. Мысалы, егер файл кітабын, сіз атап басындағы әрбір тарауда. Бар 36 белгілері қол жетімді VI. Олар белгіленеді пайдалана отырып, не әріптер А-дан Z немесе сандар 0-ден 9-ға дейін (26 әріп 10 сан береді 36 түрлі маркадағы). Қою белгісін командамен орындалады mchar командалық режимде, онда символы болып табылады әріптер А-дан Я-ға дейін немесе сандар 0-ден 9-ға дейін. Көшу ұстанымдарын көтеруді көмегімен жүзеге асырылады "жартылай кокс. Сіз сондай-ак

көшу алдыңғы белгіде бере отырып, " символы. Мысалы, егер сіз қойыңыз белгісін 'Z' басында 2-тарауының, және сіз жарты жолда арқылы 5-тарауында басу арқылы 'г' ауыстырады сіз басында 2-тарауының және 'г' қайтарады сізді нүктесі жарты жолда арқылы 5-тарауында онда сіз іске қосу алдында 'Z'.

Егер курсор орналасқан бөлгіш пайдаланылатын ++ немесе программалау тілі C / CJava, мысалы, {}, (, [немесе], онда басу % сіз басқа бөлуші белгісімен айқындалады. Мысалы, егер сіз {ол функциясын іске қосады, басып % жеткізеді сізді} жабады, ол, осы опцияны таңдаңыз. Бұл пайдалы редакциялау.

Кесте 9.4 көрсетеді әр түрлі қайталау және жою опциялары. Көптеген команданың орындалуы мүмкін бірнеше рет, егер пернесін басу арқылы нөмірімен. \$j, мысалы, орындайтын болады 'J' басулардың пернелер бес рет, яғни, ол меңзерді жылжытуға бес жолдарының төмен. Ерекшелік кезде саны алдында G әрпі, бұл жағдайда курсор басында осы жолдан; мысалы, 5G ауыстырады меңзерді жолдың басы 5. Әлбетте, кез-келген мәтіндік редактор / мәтіндік процессор талап етеді іскерліктер қиып алу және кірістіру немесе авторлық құқық және пастасы. Ретінде графикалық интерфейсі негізінде мәтіндік процессор, идеясы, таңдау реті Бойынша белгілерді, содан кейін қиып алу немесе көшіру және оларды буфер. Буферінен, олар алынуы мүмкін. Мәтіндік мәтіндік процессорлар, білдіру үшін пасталар деп жиі атайды рывком жіберу (жұлқып мәтін кері буферінен). Кесте 9.5 құрамында команда үшін кесу, көшіру және жабыстыру.

Іске қосу кезінде VI, сіз, не файлды ашуға немесе жана буфер. Содан кейін сіз редактируете VI сессиясының, сіз қажет команда файлдар. Файл барлық командалар басталады,':', - кестесінде көрсетілгендей 9.6. Мынаған назар аударыңыз, бұл олардың кейбіреулері талап ететін файлдың аты, енгізгіңіз кейін командалар: ш аты, сақтау үшін мазмұн буфер файл атымен байланысты. Онда бос орын командалары арасында) және аты бар.

9.7-кесте құрамында бірнеше жұмсалды, аралас VI команда.

Әрине, жалғыз жолы шынымен танысу VI арқылы тәжірибеге. Ескере отырып, бірнеше шетелдік басулардың пернелер басқару үшін қажетті VI, сіз күмән бар сіз бір кездері немесе оны пайдалану. Дегенмен, ретінде жүйелік әкімші немесе сценарий қабығының жазушы, сіз, бәлкім, керек болады VI (немесе Emacs) жиі. Бұрын да айтылғандай, сол пайдаланушылар Linux оқып-үйренуде VI-алғашқы, әдетте, пайдаланады VI. Сіз көресіз редакторында Emacs, көптеген командалар, бәлкім, сәл есте сақтау оңай.

9.4-кесте Қайталау және жою командалары

u	Алдыңғы команданы жою (бірнеше командаларды жою үшін бірнеше рет жою командасы қолдана алады)
.	Соңғы өзгертілген команданы қайталау
n.	Соңғы өзгертілген команданы n рет қайталау ондағы n, 5 сияқты сан
nk	Команданы k n рет қайталау ондағы k n пернелері ондағы n, 5j 5dd сияқты символдар

9.5-кесте жою, көшіру, қою, және өзгерту командалары

x	Келесі символды өшіріңіз
nx	Келесі n символдарын жою ондағы n - сан
dw	Келесі сөзді жою
ndw	Келесі n сөздерін жою ондағы n - сан
dd	Осы жолды жою
ndd	Ағымдағы жолдан бастап келесі n жолдарды жою ондағы n – сан (сондай-ақ dnd)
D	Курсордан бастаған жолды аяғына дейін
yy	Ағымдағы жолды буферге көшіру
puu	Ағымдағы жолдан бастап келесі n жолдарды көшіру ондағы n – сан (сондай-ақ упу)
p	Буферде сақталған қандай да бір жолды ағымдағы жолға қойыңыз
j	Келесі жолақты аяқ жақтағы жолаққа біріктіру (жолақтың аяқ жағына ауыстыру)
cw	Қолданушы escape пернесін басқанға дейін ағымдағы сөзді мәтінді ауыстыру арқылы өзгерту
C	Қолданушы escape пернесін басқанға дейін ағымдағы жолақтағы барлық сөздерді өзгерту

9.6-кесте файл командалары

:w<enter>	Файлды сақтау
:w name <enter>	Файлды атпен сақтау
:w name <enter>	Файлдың атын мазмұнына қарай қайта жазу
:n,m name <enter>	m арқылы n жолақтарын сақтау
:q <enter>	vi шығу
:q! <enter>	сақтаусыз vi шығу
:wq<enter> or :x <enter>	Сайлды сақтап vi шығу
:r name <enter>	Файл атын ағымдағы жолға енгізу (vi сессиясы жабық болса файлды ашу)

9.7-кесте басқа командалар

:h	Vi help файлын көру
Ctrl+T	Help файлынан өз файлыңызға қайту
!:command <enter>	Командаларды орындау (егер сіз командалар пернелерін ұмытып бірақ команданың атын білген кездепайдалы)
:=	Ағымдағы жолақтар санын экранның астында көрсету
:=	толық жолақтар санын экранның астында көрсету
^g	Ағымдағы және толық жолақтар санын экранның астында көрсету

Emacs Редакторы

Emacs (ол білдіреді редакторы Макростар) білдіреді мәтіндік редакторы, VI. Дегенмен, VI үш режимін (кірістіру, ауыстыру командасы), Emacs тек бір ғана режим (вставка). Команда беріледі пайдалана отырып немесе бақылау немесе қашуға үйлесімде. Мысалы, жылжыту үшін бір жолды алға, сіз енгізуді бақылау + п ('N' келесі). Көптеген командалар пайдаланады хат, ол болып табылады бірінші әрпі команда, және, осылайша, Emacs мүмкін оңай оқуға қарағанда,

VI. Emacs жиі болып табылады артықшылықты таңдау бағдарламашылар, себебі сәрсенбі Emacs мүмкін, егер жүйеңіз дұрыс, нұсқауларды орындаңыз. Мысалы, Лиспа тілі жиі орнатылған Emacs және сондықтан программист мүмкін редакциялау Lisp коды, содан кейін орындауға және барлық оның ішінен Emacs.

Іске қосу Emacs шығарады бастапқы "жарқырау" терезесі (сурет 9.5 А). Сіз бастағанда редакциялау (теру таңбалар немесе жүгіргі), бейнелеу Emacs ауыстырылады бос аймақты бетке (үш жол мәтін қоспағанда, см-Сурет 9.5,). Байқайсыз суретте 9.5 бар GUI команда қатысты. Егер сіз Emacs ішінде графикалық интерфейсті операциялық жүйе, сіз бақылауға Emacs арқылы тышқан (сондай-ақ арқылы басу пернелер). Дегенмен, егер сіз ашқыңыз Emacs бірі сеанс негізінде мәтінді (мысалы, егер сіз SSH компьютерге Linux), онда сіз пайдалана аласыз тек пернелерді басу.

"Emacs, өңдеу кезінде буфер. Буфер білдіреді, оның қамтитын мәтін, бұл сеанс, сіз қазіргі уақытта редакциялау. "Emacs, сіз көп буфер ашық. Бұл редакциялауға мүмкіндік береді бірнеше құжаттарды бір уақытта қиып алу / көшіру және кірістіру, олардың арасындағы. Бар мүлдем екі буфер, көрінетін кез келген сәтте: негізгі буфер (үлкен терезе кез кел-

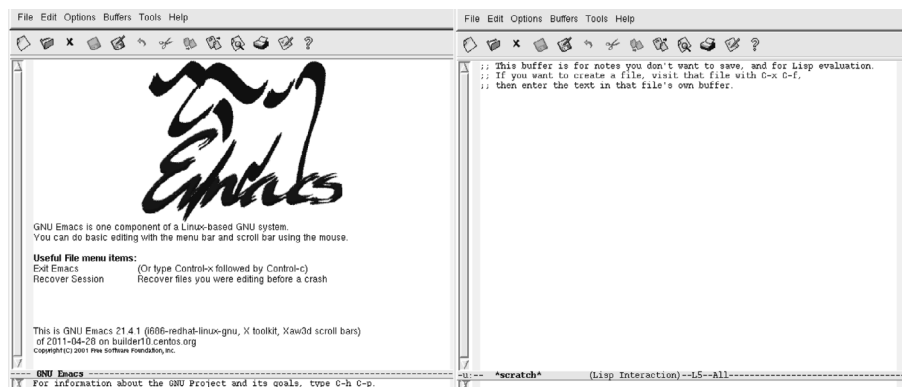
ген бөлігін сурет 9.5) және минибуфер (төменгі бөлігі терезе кез келген бөлігін сурет 9.5). Минибуфер бұл орын, онда сіз енгізуге команда, файл аттары, сондай-ақ сұрақтарға жауаптар (мысалы, "файлды Сақтау үшін керек?"). Сіз шынымен ашуға екі аралық уақытта (бұл жағдайда, үлкен терезе екіге бөлінуі), және әрбір жеке буфере, жалғастыра аласыз оны бөлуге екі буфер. Біз талқылаймыз бұл туралы кейінірек.

Emacs, сіз жетеді оң жағында буфер ретінде және кез-келген мәтіндік процессор; Дегенмен, сіз көре аласыз, бұл желі ұзартылады, өйткені жолдың соңына белгіленеді символдар

Егер сіз осындай желісі және пайдалану команда көшу үшін аяғына дейін жолдың, меңзерді жылжиды соңына қарай кеңейтілген желісі.

Мысалы, егер сізде:

Партия ботаников жоғары кабинеттерде бүкіл әлемде белгілі, қолданысқа енгізілсін т \ ОСТ жағымсыз бөлігінде заңнаманы ұтып дауыс (немесе, сол жерлерде олар d \ on't дауыс бере алады бақылау үшін жағымсыз нысандары жаппай мінез-құлық).



С + F жылжытуға меңзерді алға 10 кеңістіктер, сондай-ақ м + 3 + 2 + п жылжытуға болады меңзерді алға 32-жолдардың. Егер сіз жетеді төменгі бөлігіне файлдың жүгіргісі емес, одан әрі қозғалу және басында.

Мәтінді енгізгенде, рәміздер жүгіргі. Сондықтан Emacs әрқашан режимінде кірістіру. Редакциялау командалары жою үшін / кесу және кірістіру-кестеде келтірілген 9.9. Мынаған назар аударыңыз, бұл команда күшін жоюға болады соңғы операцияны редакциялау, жұмыс қозғалысы. Мысалы, егер сіз жойыңыз символы, күшін қалпына келтіреді сипаты. Егер жою символдар, жүгіргіні алға жылжытыңыз және болдырмау пәрменін, содан кейін жойылады символы қалпына келтіріледі, ал

жылжиды кері бұрынғы орын.

Назар аударыңыз, мәтін қосылады буферіне көшіру. Аты болғанша емес болады басқа командалар. Мысалы, егер сіз алып тастау бес жолдарының қатарынан, барлық бес жолдарының болады көшірілуі буферіне көшіру. Аты. Егер сіз алып тастау жолдың типі, C + N, өту үшін келесі жолға және жойыңыз бұл жолға тек соңғы жолы сақталады буфере көшіру. Аты, өйткені осы екі удалениями, сіз басқа команда. Дәйекті делеция, бірақ қосу буферіне көшіру. Аты.

құжат. Назар аударыңыз, егер сіз ешқашан мазмұнды буфер, Emacs болады сақтаусыз шығу. Егер сіз сақтап мазмұнды буфер файл, бірақ өзгерістер емес, сақтап қалды, бұл өзгерістер, сіз содан кейін сұрау керек файлды сақтау астында қазіргі атымен немесе жаңа аты шығуы немесе сақтау.

Команда t + x шығарады ақпарат жазылып тұрады шағын буферде. Бұл жерден сіз мына әмірді теру түймесін басыңыз іске қосу. Бар көп командалар саны. Егер сіз сенбесеңіз, болашақ, сіз пайдалана аласыз нұсқасын Emacs 'аяқталғаннан қойындылар. Теруді бастаңыз аты-жөні команда, бос орын пернесін басыңыз. Бұл тізім буфере төменгі жартысында терезенің барлық командалар сәйкес келеді. Кесте 9.11, онда талқылануда ашу, арасында ауыстыруға және буферде. Назар аударыңыз, бос орын аяқтау үшін, сондай-ақ пайдаланылуы мүмкін аяқтау үшін файл атауы. Мысалы, егер сіз қаласаңыз, файлды ашу үшін, сіз таспа + x C + F. содан Кейін сіз шақыру минибуфере енгізу файлының аты. Сіз енгізіңіз жолы / Home / foxr / citl30 / бірақ есімде жоқ файл. Басқан кезде бос орын болады, оны туғызатын барлық тиісті элементтерін, бұл жағдайда, барлық файлдарды осы тізімдемеде. Тек кейбір команда болады енгізілсін минибуферде:

9.8-кесте курсорды жылжыту командалары

c+n	Курсорды бір жолаққа төмен түсіру (келесі)
c+p	Курсорды бір жолаққа көтеру (алдыңғы)
c+f	Курсорды бір орынға алдыға қарай жылжыту
c+b	Курсорды бір орынға артқа қарай жылжыту
c+a	Курсорды жолдың басына жылжыту
c+e	Курсорды жолдың аяғына жылжыту
c+v	Курсорды экранның аяқ жағына жылжыту
m+v	Курсорды экранның жоғарғы жағына жылжыту
m+f	Курсорды бір сөзге алдыға қарай (келесі бетке)жылжыту
m+b	Курсорды артқа қарай сөздің басына (немесе алдыңғы сөзге) жылжыту

m+a	Курсорды алдыңғы сөйлемнің бас жағына жылжыту
m+e	Курсорды келесі сөйлемнің бас жағына жылжыту
m+<	Курсорды құжаттың бас жағына жылжыту
m+>	Курсорды құжаттың аяқ жағына жылжыту
m+g n	Курсорды n жолағына жылжыту ондағы n - сан
m+space	Ағымдағы орынды белгілеңіз
c+x+ c+x	Курсорды ағымдағы белгіленген орынға жылжыту (бұл сіздің орныңызды белгеленген орынға жылжытады)
c+s	Іздеуді енгізу үшін минибуферді қою, алдыға қарай іздеу (әр c+s) келесі команданы жасағанға дейін келесі команданы іздеу
c+g	Артқа қарай ауысуды іздеу

C + F жылжытуға меңзерді алға 10 кеңістіктер, сондай-ақ m + 3 + 2 + n жылжытуға болады меңзерді алға 32-жолдардың. Егер сіз жетеді төменгі бөлігіне файлдың жүгіргісі емес, одан әрі қозғалу және басында.

Мәтінді енгізгенде, рәміздер жүгіргі. Сондықтан Emacs әрқашан режимінде кірістіру. Редакциялау командалары жою үшін / кесу және кірістіру-кестеде келтірілген 9.9. Мынаған назар аударыңыз, бұл команда күшін жоюға болады соңғы операцияны редакциялау, жұмыс қозғалысы. Мысалы, егер сіз жойыңыз символы, күшін қалпына келтіреді сипаты. Егер жою символдар, жүгіргіні алға жылжытыңыз және болдырмау пәрменін, содан кейін жойылады символы қалпына келтіріледі, ал жылжиды кері бұрынғы орын.

Назар аударыңыз, мәтін қосылады буферіне көшіру.Аты болғанша емес болады басқа командалар. Мысалы, егер сіз алып тастау бес жолдарының қатарынан, барлық бес жолдарының болады көшірілуі буферіне көшіру.Аты. Егер сіз алып тастау жолдың типі, C + N, өту үшін келесі жолға және жойыңыз бұл жолға тек соңғы жолы сақталады буфере көшіру.Аты, өйткені осы екі удалениями, сіз басқа команда. Дәйекті делеции, бірақ косу буферіне көшіру.Аты.

Мысалы, қарастырайық, келесі жол:

Қоңыр сиыр күміс қасықтан секірді

Болжаймыз, не курсор орналасқан қазіргі уақытта "V" сөз "бақылау". Егер сіз m + D, ол жояды "тігі". Егер сіз жасасаңыз тағы бір t + d, ол жояды "The". Егер сіз c + y, "Ver" қайтадан пайда болады. Егер сор орналасқан сол - сол жерде, және сіз m + <Backspace> екі рет, ол жояды "секірді OV", бірақ "э-э". Егер сіз қазір C + F, кейіннен t + D, бұл жояды "г", "эр". Егер сіз орындаңыз осы c + y, барлық нәрсе болып табылады рывком "г", өйткені бұрын "секірді OV" енді буферінен. Файлдық операциялар кестеде келтірілген 9.10. Минibuфер пайдаланылады сұрау

үшін пайдаланушы аты үшін пайдалану кезінде сақтауға, ашуға және сіз келсе, Emacs шығу.

9.9-кесте Жөндеу командалары

c+x u	Жою (c+ және c+)
c+d	Келесі символды жою (сондай-ақ жою пернесі)
m+d	Келесі сөзді жою (курсор тұрған жерден)
c+k	Толық жолақты жою (курсор тұрған жерден) (сондай-ақ m+k)
<backspace>	Алдыңғы символды жою
m+<backspace>	Курсордан соңынан сөзді басына дейін жою
m+w	Толық белгіленген жерді соңында белгіленген түктен бастап сақтау (көшіру)
c+w	Толық белгіленген жерді соңында белгіленген түктен бастап жою (кесу)
c+y	Белгіленген мәтінді буферге сақтау, сақталған белгіленген мәтінді буферден кою (жою және c+d баска жоғарыдағы енгізілген символдарды буферге сақтау)
m+y	c+y сияқты ол буфердегі мәтінді алмастырады
c+t	Ағымдағы және алдындағы символды ауыстыру
m+u, m+l	Жоғарғы/төмен жағдайлар барлық әріптер курсордан сөздің соңына дейін
m+c	Ағымдағы әріпті бас әріппен жазу
m+%	Іздеу және орнын алмастыру - іздеу және орын алмастыру туралы хабарлама болады, орын алмастыру (y) , өткізіп жіберу(n), шығу (q) орын алмастыру(!) және шығу(.) туралы сұралады.

Команда Кесте 9.10 Файл

c + x c + s	Файлды ағымдағы атаумен сақтау
c + c + x	Сақтау (үлгіде көрсетілген атаумен сақтау)
+ x C + F	Жаңа файлды жаңа буферде ашу
c + x C + C	Emacs-тан шығу (егер мәтін сақталмаса бірақ файл атауы болса, шығу кезінде мәтінді сақтау керектігі сұралады)

күжат. Назар аударыңыз, егер сіз ешқашан мазмұнды буфер, Emacs болады сақтаусыз шығу. Егер сіз сақтап мазмұнды буфер файл, бірақ өзгерістер емес, сақтап қалды, бұл өзгерістер, сіз содан кейін сұрау керек файлды сақтау астында қазіргі атымен немесе жаңа аты шығуы немесе сақтау.

Команда t + x шығарады ақпарат жазылып тұрады шағын буфере. Бұл жерден сіз мына әмірді теру түймесін басыңыз іске қосу. Бар көп командалар саны. Егер сіз сенбесеңіз, болашақ, сіз пайдалана аласыз нұсқа-

сын Emacs 'аяқталғаннан қойындылар. Теруді бастаңыз аты-жөні команда, бос орын пернесін басыңыз. Бұл тізім буфере төменгі жартысында терезенің барлық командалар сәйкес келеді. Кесте 9.11, онда талқылануда ашу, арасында ауыстыруға және буферде. Назар аударыңыз, бос орын аяқтау үшін, сондай-ақ пайдаланылуы мүмкін аяқтау үшін файл атауы. Мысалы, егер сіз қаласаңыз, файлды ашу үшін, сіз таспа + x C + F. содан Кейін сіз шақыру минибуфере енгізу файлының аты. Сіз енгізіңіз жолы / Home / foxr / cit130 / бірақ есімде жоқ файл. Басқан кезде бос орын болады, оны туғызатын барлық тиісті элементтерін, бұл жағдайда, барлық файлдарды осы тізімдемеде. Тек кейбір команда болады енгізілсін минибуферде:

- қосу файлы
- автоматты түрде сақтау режимін
- калькулятор
- күнтізбе
- ккт-область, бас әріптермен сөз
- Чек-дөңгелек жақшаға
- файлды көшіру
- саны сызықтар, беттердің саны желілер-облысы

Кесте 9.11 Буфер операциялары

c + x k	Ағымдағы буферді жабу (егер сақталмаған материал бар болса, оны бірінші сақтаңыз деген сұраулар болады)
c+x b	Ағымдағы буферден ең соңғы буферге ауысу
c + x 2	Терезенің жарты өлшемін алатын екінші буферді көру үшін екінші терезені ашу (ағымдағы терезе астынан), (ескерту: жартылай ашылған терезенің ішінен ашқан кезде ширек екі терезе және т. б береді.)
c + x 1	Екінші терезені бір терезеге жабу
c + x O	Минибуферді қоса алғанда бір буферден екінші буферге ауысу

- Отог желісі
- кірістірме-файл
- сақтау-буфер
- қабығы (ашады қабығы Linux)
- Undo

Бар бірқатар бағдарламалар, мүмкін іске қосылды ішінде Emacs. Оларға

дәрігер (жасанды интеллект), ханой (Мұнара Ханоя), фаза-оф-ай (көрсету үшін сізге алдағы фазасының) және vestr. Сіз орындауға көмегімен бағдарламаны $M + x$ аты ретінде $m + x$ дәрігердің <кіруге> іске қосу үшін бағдарламаның докторы. Ішінде Emacs, сіз орындауға команда Linux. Бар бірнеше режимдерін тексеру орнатылған, қол жетімді көмегімен не Aspell немесе Ispell. Flyspell (Aspell) бірінші орынға қояды барлық сөздер қателермен буфере. Сіз сондай-ақ пайдалану Flyspell режимі бағдарламалау, ол тек көрсетеді сөздер қатемен ескертулер (мысалы, табуға болады барлық орфографиялық қателер, ескертулер Java бағдарламасының). Ispell үшін пайдаланылады қалған бөлігін режимдерін. Ispell тексере алады бөлінген сөзді немесе аяқтау мүмкін сөз, немесе білдіруі мүмкін тексеру ағымдағы, аралық немесе ағымдық аймақ. "Ispell, терезе бөлінеді екі буфер тізімімен деген сөздерден үшін деген қателер туындайтын төменгі буфере. Сіз сондай-ақ қосу сөздерді сөздік Ispell тілек. Іске қосу үшін не емлені пайдаланыңыз $M + x$ <команда>, <команда> бірі болып табылады мынадай іс-әрекеттерді:

- Flyspell-режимі
- Flyspell-бағ режимі
- Ispell
- Ispell-буфер
- Ispell-аймақ
- Ispell-хабарлама
- Ispell-өзгерту-сөздік (өзгертуге мүмкіндік береді сіздің сөздік Ispell)

Сіз сондай-ақ пайдалануға $M + \$$ үшін емлені ағымдағы сөздер және $M + x$ <қойындысы><Esc>

<Tab> аяқтау үшін ағымдағы сөз. Пайдалану кезінде Ispell, егер сөз дұрыс таңдалған, сіз жасай аласыз кез келген осы іс-әрекеттер:

- <бос орын> -елемеу деген сөз (ол әлі де болып саналады орфографиялық қате, жай ғана менсінбей келген уақытта)
- г сөз <оралу> -деген бөлінген сөз деген
- сан-деген сөз сөзімен, таңдап алынған сандар енгізілген тізімінен
- а-сөз ретінде
- менің сұғындыр-бұл сөз сөздікте
- ? -show толық тізім параметрлер, пайдаланылуы мүмкін Ispell

Бір соңғы ерекшелігі Emacs мүмкіндігі болып табылады анықтау және

қайта қоңырау шалу макростарды (бұл сондай-ақ қол жетімді VI болмаса да, бұрын талқыланды). Макрос бағдарламасын білдіреді, ол анықтауға болады ретінде дәйектілігі басулардың пернелер орындалатын бір реттен артық. Сіз анықтайсыз макрос, одан кейін шақыру, оған әр жолы, сіз оны қажет болған кезде. Іске қосу үшін немесе жояды енгізіңіз с + х (кейіннен макродеңгейде басулардың пернелер бұдан кейін с + х). Орындау үшін енгізілген немесе жояды үлгідегі + х е

Мысал қарастырайық. Елестетіп көріңізші, сіз бар мәтінді, сіз қоюға болады бұл мәтін HTML-кесте. Текст қазіргі уақытта ретінде анықталады # Ch. # п # әр # болып табылады саны. Сіз қоюға болады бұл HTML-кестенің әр жолы HTML кестелер сәйкес келетін жол мәтіндік кесте). Бірінші # болады орналастырылуы бір бағанға, Ch. # Екінші бағанада, р. # үшінші бағанада көрсетілген. HTML-кесте басталады <кесте> аяқталады </ TABLE> және әрбір жолы тұрады: <TR><TD> 1-баған </ TD><TD> 2-баған </ TD><TD> 3-баған </ TD></ TR>. Сондықтан бізге қосу әрбір жолында <TR><TD> алдында бірінші #, </ TD><TD> алдында Ch. #, </ TD><TD> алдында р #, </ TD></ TR> соңына дейін жолдар. Орындау үшін біздің міндет жоғарыда, біз жасауға болады мынадай:

- Тінтуірді жылжыту, бірінші жолда кестенің түрі <кесте><Enter> және типі + х (
- Түрі + а <TR><TD> ауыстырады -мензерді жолдың басы қосады мәтін <TR><TD> басында
- Түрі м + ж-бұл мензерді ауыстырады өтті бірінші сөз (бұл жағдайда, бірінші саны)
- Түрі </ TD><TD> -соңы бірінші баған мен екінші бастау
- Түрі + s. + s. + Ь + Ь-бұл мензерді ауыстырады алдында р.
- Түрі </ TD><TD> -соңына екінші бағанды бастау үшінші
- Тип с + е </ TD></ TR> - пресс-центр, көшу соңына қарай жолдың соңына бағанның және жолдың
- Түрі + -п-ауысу үшін, келесі жолда
- Түрі + х) -соңына немесе жояды

Егер 15-жолдар үшін кестелер, енді біз С + 14 + х е (орындауға макрос 14 рет) және, сайып келгенде, түрі </ кесте>. Сіз сақтап, макростар және оларға аттары, сондықтан сіз анықтау көптеген макростар және еске макрос қажет. Кейінірек, сіз редакциялауға макрос аты және қосу оған немесе өзгерістер.

Ақырында, Emacs мүмкін тұрып түсіруге абзал. Егер сіз жоғалтып алсаңыз, сіз шақыртуға анықтама файлын жаңа буфере. Тәрбиелеуге

көмек үлгідегі $+ H + H$, немесе егер сіз көмектесе командасы бойынша нақты командасы, түрі $C + H$. Егер сіз сенімсіз болсаңыз, дұрыс команда енгізіңіз бірінші әріп немесе әріптер Ком - сұраныс. Сондықтан да VI, үздік тәсілі білуге Emacs болып табылады практика.

10. ТҰРАҚТЫ ӨРНЕК

9 тарау Linux арналған болатын. 10 тарауда Linux қатысты тақырып – тұрақты өрнекке арналады. Тұрақты өрнек пайдаланушылар және Linux әкімшілігі үшін маңызды құрал болып табылады. Пайдаланушы тұрақты өрнектің көмегімен мәтінді файлдар арқылы қандай да бір нақты жолдарды ғана емес, сонымен қатар айқындалған қызықты дайын үлгіге сәйкес келетін жолдарды да іздестіруді жүзеге асыруы мүмкін. Linux берілген тұрақты өрнек бойынша іздеу тапсырмасын орындайтын grep утилитасын ұсынады. Бұл тарауда тұрақты өрнек көптеген мысалдармен және grep утилитасын қарау арқылы қарастырылады. Тұрақты өрнекті зерттеу кезінде олардың табиғатынан күрделілігіне байланысты оқырман шеберліктің оларға тек қана үлкен жұмыс тәжірибесі арқылы келетіндігін ұмытпауы тиіс, сондықтан бұл тарауды оқып түсіну қиын болуы мүмкін. Оқырманға осы мысалдардың көбін байқап көру ұсынылады. Сондай-ақ, бұл тарауда Bash джокер-таңбалардың қалай қолдануға болатыны қарастырылады.

Осы тарауды оқытудың мақсаты төмендегідей болып табылады:

- Тұрақты өрнекті және олардың неліктен пайдалы екендігін сипаттау.
- Тұрақты өрнектің әрбір метатаңбасын суреттеу.
- Тұрақты өрнектің көптеген мысалдарын қамтамасыз ету.
- Grep утилитасын тексеру.
- Bash джокер-таңбаларын пайдалануды сипаттау және тұрақты өрнектен айырмашылығын көрсету.
- ls және grep қайта бағыттау арқылы біріктіру.

ls ғана қолданып, кейіннен 0s айналатын таңбалар жолдарын қарастырайық, мәселен, 111000,100 және 10000. Осындай үлгіні белгілеу үшін тұрақты өрнекті қолдануға болады. Жазылған тұрақты өрнекті жолдар

тобымен салыстыруға әрі дайын үлгіге сәйкес келетін өрнектерді ғана қалдыруға болады. Тұрақты өрнек – бұл өзінде әріптік таңбаларды (мысалы, 0 не 1), метатаңбаларды және бірнеше нұсқаларды ұсынатын таңбаларды үйлестіретін жол. Сіз метатаңбалардың көмегі арқылы мысалы, аталмыш таңба немесе таңбалар жиынтығы «кез келген мөлшерге» не «тым болмағанда бір рет» ұғымына сәйкес келуін не болмаса «кез келген таңбаның» үйлесімділігін көрсететін таңбалар тізімін айқындай аласыз.

Тұрақты өрнек пайдаланушыға немесе жүйелік әкімшілік, файлдарда сақталатын файлдарды немесе элементтерді іздеу үшін пайдалы болуы мүмкін. Бұл тарауда тұрақты өрнекті анықтау және оларды Linux пайдалануды қарастырамыз. Тұрақты өрнек соншалықты пайдалы деп саналғандықтан, Linux пайдаланушылары үшін маңызды құрал болып табылатын `grep` атты біріктірілген утилитға ие болды. Джокер-таңбалары, тұрақты өрнек түрлері Linux қолжетімді болғанымен, олар тұрақты өрнектің қолдану жолдарына байланысты әр түрлі жеткізіледі. Тұрақты өрнектер бағдарламалаудың бірнеше тілдерінде құрылған, сондықтан Linux, сонын ішінде Perl бағдарламасында кеңінен қолданысқа ие.

Тұрақты өрнекті зерттеу бізге не үшін керектігін түсіндіру үшін екі қарапайым мысал қарастырайық. Біріншіден, сіз, пайдаланушы ретінде тізімдемедегі суреттерге қол жеткізе аласыз. Арасында JPG, GIF, PNG, TIFF форматтағы суреттер бар. Сіз барлығын Tiff форматқа аударғыңыз келеді. Дегенмен, сіз басқа қолданушылардың оларды қалай атағанына - `.tiff`, `.tif`, `.TIFF`, `.TIF`, `.Tiff` немесе `.Tif` кеңейту, сенімді емессіз. Мүмкін деген барлық кеңейтулерді атап шығу үшін алты түрлі `Ls` нұсқауларды немесе тіптен бір ғана `Ls` нұсқаулықтың орнына тұрақты өрнекті қолдана аласыз. Екіншіден, жүйелік әкімшілік ретінде, кейбір қатты бағдарламаланған IP-мекенжайды өзгерту үшін сіз IP-мекенжайы бар барлық файлдарға арналған (`say /etc`) тізімдемені іздеу керексіз, бірақ қай файлдарды тексеру керек екені есіңізде жоқ. Тұрақты өрнек `#. #. #. #` жол түріне сәйкестендіріліп құрылады, бұл жердегі әр `#` белгі – 0-ден 255 диапазон аралығындағы мағына. Осындай құрылым кезінде, тұрақты өрнекті және `grep` пайдалану арқылы, ондаған немесе жүздеген файлдарды зерттеудің орнына, сіз барлық сәйкестікті бір команда беру арқылы көре аласыз.

Бұл тарауда біз алдымен тұрақты өрнек үшін керекті метатаңбаларды қарастырайық. Біз тұрақты өрнектің ондаған мысалын және олардың немен сәйкес келетіндігін қарастырамыз. Содан кейін, біз кейбір таңба-

лардың Bash интерпретаторы арқылы джокер-таңбалары ретінде қалай қолданылатынын қарастырамыз. * белгісі gper утилитасында тұрақты өрнек ретінде және Bash интерпретаторы оны ls нұсқаулығында қолданғанда түрлі мағынаға ие болғандықтан бұл түсініспеушілікке әкелуі мүмкін. Және, ақыр соңында, біз gper утилитасын және оны қалай пайдалану керектігін қарастырамыз. Тұрақты өрнекті дұрыс қолдану мәселе туындатуы мүмкін. Көптеген жағдайларда, олардың мәні айқын болғанымен олармен таныс емес пайдаланушыларды шатастыру мүмкін. Оларды қолдану кезінде шыдамдылық танытсаңыз, соңында сіз тіпті ләззат аласыз.

МЕТАТАҢБАЛАР

Үлгіде (шаблонда) параметрлерді сипаттау үшін қолданылатын көптеген таңбалар бар. Олар метатаңбалар ретінде белгілі. Кез келген тұрақты өрнек әріптік таңбалар мен метатаңбаларды қамтитын болады (бірақ тұрақты өрнек метатаңбаларды қажет етпейді). * метатаңбасы «алдыңғы 0 таңбасына немесе бірден көпке сәйкес келеді» дегенді білдіреді; мысалы, a* «нөл немесе көп A» дегенді білдіреді. 1010 тұрақты өрнегі тек 1010-ға сәйкес келеді, себебі мұнда метатаңбалар жоқ. Әдетте біз тұрақты өрнектер бір нақты жолдан көбіне сәйкес келгенін қалайтын болсақ, біз әрқашан метатаңбаларды қолданатын боламыз. 10.1 кесте метатаңбалар тізімін қамтиды. Әрі қарай тарауда біз олардың әрқайсысын қарастырамыз.

Біз ең негізгі таңбалардан бастаймыз: * және +. Олардың кез келгенін қолдану үшін, алдымен салыстыру үшін таңбаны көрсету қажет, сосын бұл 0 таңбасын немесе (*) бір реттен көп немесе 1 немесе (+) бір реттен көп көруді күтетінімізді көрсету үшін * немесе + метатаңбасын таңбадан кейін орналастыру қажет.

10.1 КЕСТЕ. Тұрақты өрнектер метатаңбалары

Метатаңба	Түсіндірме
*	Алдыңғы таңбаға сәйкес келеді, егер ол 0 немесе бір реттен көп пайда болса
+	Алдыңғы таңбаға сәйкес келеді, егер ол 0 немесе бір реттен көп пайда болса
?	Алдыңғы таңбаға сәйкес келеді, егер ол 1 немесе 1 реттен көп пайда болса
.	Кез келген таңбамен сәйкес келеді
^	Сәйкес келеді, егер бұл өрнек жолды бастаса
\$	Сәйкес келеді, егер бұл өрнек жолды аяқтаса

[chars]	Сәйкес келеді, егер өрнек []-дегі кез келген таңбаларды қамтыса
[chari-charj]	Сәйкес келеді, егер өрнек char _j -ден char _j -ге дейінгі ауқымдағы кез келген таңбаларды қамтыса (мысалы, A-Z, 0-9)
[[:class:]]	[] балама формасы, мұнда: class: альфа (әліпбилік), сан, арнайы өрнек (әріптік немесе сандық), пунктуатор, кеңістік, жоғарғы, төменгі секілді бірнеше санаттардың біреуі болуы мүмкін
[^chars]	Сәйкес келеді, егер өрнек []-дегі кез келген таңбаларды қамтымаса
\	Келесі таңба бірігуі және метатаңбаның мәнінен қашу үшін қолдануы тиіс, мысалы, \\$ "\$-ке сәйкес келу" дегенді білдіреді
{n}	Сәйкес келеді, егер жол алдыңғы таңбаның n жүйелі болуын қамтыса
{n,m}	Сәйкес келеді, егер жол n және m арасындағы алдыңғы таңбаның жүйелі болуын қамтыса
{n,}	Сәйкес келеді, егер жол кем дегенде n алдыңғы таңбаның жүйелі болуын қамтыса
{,m}	Сәйкес келеді, егер жол m-нен көп емес алдыңғы таңбаның жүйелі болуын қамтыса
	Осы жолдардың кез келгеніне сәйкес келеді ("НЕМЕСЕ")
(...)	Элементтер барлық жүйелілікке сәйкес келетін топ ретінде қарастырылады

Мысалы, $0 * 1 *$ артында нөл немесе 1s-ден көп болатын нөлден немесе 0s-ден көп кез келген жолға сәйкес келеді. Бұл тұрақты өрнек мына жолдарға сәйкес келетін болады: 01, 00011111, 1, 00000, 0000000001 және бос жол. Бос жол – таңбалары жоқ жол. Бұл өрнек бос жолға сәйкес келеді, себебі $*$ таңбасы 0 сәйкестік үшін қолданылуы мүмкін, сондықтан $0 * 1 *$ 0s жоқ және 1s жоқ жолына сәйкес келеді. Бұл тұрақты өрнектің, мысалы келесі мәндердің ешқайсысына сәйкес келмейді: 10,00001110, 0001112, 00a000 немесе әліпби. Бірінші және екінші жағдайларда 0 таңбасы 1-ден кейін келеді. Келесі үш жағдайда жолда 0 және 1-ден басқа таңбалар бар.

$0 + 1$ тұрақты өрнегі кем дегенде бір 0 және бір 1 болу керектігін көрсетеді. Осылайша, бұл тұрақты өрнек бос жолмен сәйкес келмейді; сонымен қатар, ол екі сандардың бірін қамтитын ешқандай жолға (мысалы, ол 0000 немесе 1-ге сәйкес келмейді) сәйкес келмейді. Бұл өрнек 01, 00011111 және 0000000001-ге сәйкес келер еді. $0 * 1 *$ секілді ол 0 немесе 1-ден басқа таңбалар бар жолға (мысалы, 0001112) сәйкес келмейді, сондай-ақ, 0 таңбасы 1-ден кейін келетін жолға да (мысалы,

00001110) сәйкес келмейді.

Әрине, біз тұрақты өрнекте * және + таңбаларын 0 * 1 + немесе 0 + 1 * секілді үйлестіре аламыз. Біз * немесе + сандық таңбаларсыз көрсете аламыз. Мысалы, 01 * артында нөл немесе көп ls болатын 0-ге сәйкес келетін болады, мысалы, 0, 01, 01111, бірақ, 1, 111, 1110 немесе 01a-ға тең емес. * және + түсіну оңай болғанымен, келесі таңбаны көрсетуге болатын кезде олардың пайдалылығы шектеулі. Біз * және + таңбалары қайталанатын таңбалардың комбинациясын көрсету үшін, оларды []-мен біріктіруге болатын кезде пайдалырақ болатындығын көреміз.

? – бұл * немесе + секілді нұсқа, бірақ ол алдыңғы таңбаға тек 0 немесе 1 рет сәйкес келетін болады. Бұл таңбаны күтуге болатын немесе болмайтын кездегі жағдайды анықтауға көмектеседі. Алайда, бұл сәйкестіктің бірнеше рет болу мүмкіндігін білдірмейді (* немесе + секілді). tiff / TIF мысалын еске түсірейік. Біз tiff немесе tif келесі түрде сәйкес келетін тұрақты өрнекті көрсете алар едік: tiff?. Бұл жағдайда біз үш бастапқы таңбаны күтеміз, tif. Алайда, соңғы таңба, f, 0 немесе 1 рет пайда болуы мүмкін. Бұл тұрақты өрнек TIFF секілді жолдарды, яғни бірыңғай бас әріптерді қанағаттандырмағанымен, осыдан бастауға болады. Енді, ?, * және + таңбаларынан таңбаны 0 немесе 1 рет, 0 немесе бір реттен көп, 1 немесе бір реттен көп көруді күтетінімізді бақылай аламыз.

* және + таңбаларына қарағанда ? таңбаны көруді күту ретіне шектеу қояды. Сондықтан, ? таңбасынан біз шынымен де сәйкес келуі мүмкін барлық комбинацияларды атап шыға алар едік. Мысалы, 0? 1? тек төрт ықтимал жолдарға ғана сәйкес келетін болады: 0, 1, 01 және бос жол. 0 * 1 * жағдайында сәйкес келуі мүмкін жолдардың шексіз саны бар, себебі, "0 немесе көп" жоғарғы шегі жоқ.

* және ? Linux пәрмендерінде ls джокер-таңбалар ретінде қолданылатынына назар аударыңыз. Bash және Джокер-таңбаларда осындай пәрмендерде олардың мәні осында келтірілген мәндерден ерекшеленетіндігін білеміз.

. (кезен) кез келген жеке таңбаға сәйкес келуі үшін қолданылуы мүмкін. Мысалы, b.t мына жолдардың кез келгеніне сәйкес келе алады: bat, bet, bit, but, bot, bbt, b2t. Ол bt, boot, or b123t жолдарына сәйкес келмейді. Метатаңбамен *, + және ? таңбаларын комбинациялауға болады. Мысалы, b.*t b-дан басталатын, сосын оның артында басқа таңбалардың кез келген саны (соның ішінде таңбаларсыз да) болатын және t-ға аяқталатын кез келген жолға сәйкес келетін болады. Осылайша, b.*t мыналармен сәйкес келеді: bat, bet, bit, but, bot, bbt, b2t, bt, boot, b123t және әрі қарай. b.+t өрнегі осындай, бірақ b мен t арасында кем дегенде бір таңба болуы керек, сондықтан оған bt жолынан басқа кез келген барлық

жолдар сәйкес келетін болады. b.?t тұрақты өрнегі b.t. сәйкес келетін bt немесе барлығына сәйкес келетін болады. Сұрау белгісі . (кезең)-ге сәйкес келеді. Сәйкесінше, . 0 немесе 1 рет қолданылады; сондықтан, бұл bt немесе b.t. сәйкес келетін тұрақты өрнекті береді. Ол b және t арасында бірден көп таңбаны қамтитын бірде бір жолға сәйкес келмейді.

Келесі метатаңба жинақ немесе тізімді беру үшін қолданылады. Ол [-ден басталады, тізімді қамтиды және]-ге аяқталады. Мысалы, [aeiou] немесе [123]. Мұндай үлгінің тізімдегі элементтердің қандай да біреуін қамтитын кез келген жолға сәйкес келетіндігінде көрінеді. Біз мысалы, bat, bet, bit, bot, және but сәйкес келуі мүмкін b[aeiou]t көрсете аламыз. Немесе біз жоғарғы немесе төменгі регистрді көрсету үшін []-ді қолдана аламыз. Мысалы, [Tt] төменгі t регистріне немесе жоғарғы T-ға сәйкес келетін болады.

Енді, бізде tif/tiff кез келген формасына сәйкестік үшін қажетті құралдар бар. Келесі тұрақты өрнек кішкентай және бас әріптердің кез келген комбинацияларын қолдана отырып, tif немесе tiff кез келген түрлеріне сәйкес келетін болады: [tT][iI][fF][fF]?. ? төртінші []-ге ғана жатады. Осылайша, ол артында бас немесе кішкентай i әрпі келетін, сосын бас немесе кішкентай f болатын және нөлге немесе жоғарғы регистрдегі бір f-ке аяқталатын, бас немесе кішкентай t әріпті кез келген жолдарға сәйкес келетін болады.

Жақшадағы элементтер саны толықтай нөмірленген тізім болуы міндетті емес. Оның орнына, а-дан г-ге дейінгі әріптер ауқымы болуы мүмкін. Қатар ауқымдағы бірінші таңбамен, дефиспен (-) және ауқымдағы соңғы таңбамен көрсетіледі. Ауқымдарға арналған рұқсат етілетін таңбалар сандар, кішкентай және бас әріптер болып табылады. Мысалы, [0-9] кез келген санды білдіреді, ал [a-g] а-дан г-ге дейінгі төменгі регистрдегі кез келген таңбаны білдіреді. Яғни, [a-g] [абвг]-ге баламалы болып келеді. Сіз тізімде көрсетілген диапазон мен таңбаларды қоса аласыз, мысалы, [б-вг-тв-хг] кішкентай әріпті дауыссыз дыбыстар тізімі болады.

Айтылған тізімге немесе ауқымға балама ретінде қос жақшалар мен сыныпты қолдануға да болады. Мысалы, [a-zA-Z] орнына сіз әліпбилік таңбалар сыныбын білдіретін [[alpha:]]-ді қолдана аласыз. 10.2 кестеде көрсетілгендей, Linux тұрақты өрнектеріне қолжетімді 12 стандартты сыныптар бар. Астын сызу таңбасын қосқанда барлық әріптік-сандық таңбалардан тұратын стандартты емес сынып [:сөз:].

[] немесе [[]] көмегімен көрсетілген тізімде жолда табылған кез келген бір таңбаға сәйкес келетін болады. Егер сіз ауқымдағы таңбалардың қандай да бір комбинациясына сәйкес келгенін қаласаңыз, онда сіз жақшалардан кейін *,+,., немесе ? қоса аласыз. Мысалы, [a-z]+ төменгі ре-

гистрдің бір немесе бірнеше әріптерді білдіреді.

Сіз қандай да бір атқа сәйкестікті тапқыңыз келеді деп қарастырыңыз. Біз аттың бірінші әрпі бас әріп болатындығын білмейміз, бірақ қалған әріптер біз төменгі регистрде болады деп күтеміз. Бірінші әріпке сәйкестік үшін біз [A-Za-z] қолданатын боламыз. Яғни, біз жоғарғы немесе төменгі регистрдің әрпін күтеміз. Сосын біз бірнеше кішкентай әріптер санын күтеміз, оны [a-z]+ деп белгілейміз. Біздің тұрақты өрнегіміз [A-Za-z][a-z]+. Біз әріптердің төменгі регистрі үшін + орнына * қолдануымыз керек пе? Бұл аттың бір әріптен тұруы мүмкіндігін күтетінімізге байланысты. Біз инициалды емес, атты күтетіндіктен, ол бірнеше әріптен тұрады деп ойлаймыз. Дегенмен, біз егер ат мысалы, J болуы мүмкін деп ойласақ, онда [A-Za-z][a-z]* қолданар алар едік.

[A-Za-z0-9] * қайсысымен сәйкес келуі мүмкін? Бұл өрнек нөлге не әріптер немесе сандардың көп санына сәйкес келетін болады. Ол сондай-ақ бос жолды (* өзіне нөлдік енуді қосқандай), кез келген бір әріпті (жоғарғы немесе төменгі регистр) не санды немесе әріптер мен сандардың кез келген үйлесімін қамтиды. Осылайша, сәйкес келетін болады: abc ABC aBc a12 1B2 12c 123456789 aaaaaa 1a2b3C4D5e

Сәйкес келмейтіні болады ма? Әріптер мен сандардан басқа таңбаларды қамтитын кез келген жол. Мысалы, a_b_c, 123!, a b c (яғни, әріптер мен бос кеңістік), сондай-ақ, a1#2B%3c*fg45.

10.2 КЕСТЕ. Тұрақты өрнектердің метатаңбалары

Сыныбы	Мәні
[[:alnum:]]	Әріптік-сандық таңба – әріптік таңба (әріп) немесе сан
[[:alpha:]]	Әліпбилік - әріп (жоғарғы немесе төменгі регистр)
[[:blank:]]	Бос орын немесе tab
[[:cntrl:]]	Кез келген басқару таңбасы
[[:digit:]]	Сан
[[:graph:]]	Кез келген көрінетін таңба
[[:lower:]]	Кішкентай әріп
[[:print:]]	Кез келген көрінетін таңба қосу бос орын
[[:punct:]]	Кез келген тыныс белгісі
[[:space:]]	Кез келген бос орын таңбасы (tab, return пернесі, бос орын, backspace)
[[:upper:]]	Бас әріп
[[:xdigit:]]	Он алтылық сан

□ көмегімен біз қандай таңбалардың сәйкес келетіндігін бақылай аламыз, бірақ олардың пайда болуын бақылай алмайтындығымызға назар аударыңыз. Егер, біз, мысалы, 'b' 'a'-дан кейін тұрғанын талап етсек, біз оларды жақшалардың екі жинағын қолдана отырып, жүйе етіп жаза-

мыз, мысалы, артында бас немесе кішкентай 'b' әрпі келетін 'a' жоғарғы немесе төменгі регистрін көрсету үшін [aA][bB]. Біз [aA]+[bB]+ (немесе + орнына * қолдана отырып) көмегімен олардың кез келген санына жол бере аламыз. Бұл егер біз әріптердің комбинациясы комбинациядан кейін келгенін қалайтын болсақ, күрделі бола түсуі мүмкін. Біз артында дауысты (e) мен одан кейін дауыссыз (и) тұратын дауыссыз (e) бар таңбалардың кез келген жолына сәйкес келетіндей тұрақты өрнекті жасағымыз келеді деп ойлаңыз. Біз келесі тұрақты өрнекті қолдана алар едік:

[b-df-hj-np-tv-z]+[aeiou]+[b-df-hj-np-tv-z]+

"1 немесе көп"? * көбірек бақылауды қолдануға болады ма және + біз қаншалықты көп сәйкестіктердің болуын ойламаған кезде сәйкес келеді, бірақ біз шектеуге ие болғанымызды қалар едік. Мысалы, бестен көп болмауы керек немесе екі сәйкестіктен кем болмауы керек. Біз қандай да бір комбинацияның көмегімен осыны орындай аламыз ба? Метатанбаның? Мысалы, "бестен көп емес" дегенді көрсету үшін, "?????" қолдана алар едік. Өкінішке орай, біз бұл жағдайда сұрау белгісін сәйкестендіре алмаймыз. Бірінші сұрау белгісі таңбаға жатады, ал қалған сұрау белгілері алдыңғы сұрау белгісіне қолданылады.

Алайда, біз сұрау белгісі келетін таңбаны ("кез келген таңбаға" арналған кезең) орналастырып, осыны бес рет қайталар едік:

.....?

Бұл тұрақты өрнек әрбір сұрау белгісін кезеңде қолданады. Және сұрау белгісі "0 немесе 1" білдіретін болғандықтан, кез келген бір, екі, үш, төрт немесе бес таңба деп айтқанмен бірдей. Бірақ біз таңбаларды осындай таңбалар болуға мәжбүрлей аламыз? [[:visible:]]?[:visible:]]?[:visible:]]?[:visible:]]?[:visible:]]?[:visible:]] қолдансақ қалай болады? Өкінішке орай, кезеңде орын алғандай, әрбір [[:visible:]] –дегі таңба кез келген көрінетін таңба бола алады, бірақ міндетті түрде қажет болатын таңба емес.

Біздің шешеміз – басқа метатанбаны қолдану, бұл жағдайда {n,m}. Мұнда n мен m, n m-нен аз оң бүтін сан болып табылады. Бұл жазба алдыңғы таңбаның n мен m арасындағы кірулерге сәйкес келетін болады дегенді білдіреді. Яғни, " тым болмағанда n-мен сәйкестік алдыңғы таңбаға қарағанда көп емес". Сіз "тым болмағанда" немесе "көп емес"-ті түсіріп тастай аласыз немесе "дәл" қамтамасыз ету үшін бір мәнді көрсете аласыз.

Мысалы, 0 {1,5} 1 * келесі кез келген 1s саны бар "бір 0s-тен беске дейін" дегенді білдіреді, ал [01] {1,5} "бірден беске дейінгі 0 және 1 комбинациясы" дегенді білдіреді. Соңғы жағдайда бізге 0s пен 1s қандай реттілікте келетіндігі бәрібір. Сәйкесінше, бұл соңғы өрнек 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, бес бүтін таңбаларға дейін

сәйкес келетін болады.

Біз {2}-ді "тым болмағанда, екі" және {5}-ті, "бестен көп емес" дегенді көрсету үшін қолдана аламыз. {n,m} қолдана отырып, біз соңғы санмен сәйкестік санын шектей аламыз. 0 {5} 1 {5} қарастыру. Бұл тек 0000011111-ге сәйкес келетін болады. Дегенмен, [01] {5} бес 0s пен 1s тұратын кез келген комбинацияға сәйкес келетін болады.

{ } зерттеу тек кеңейтілген тұрақты өрнекті қолданған кезде ғана қолжетімді екендігін айта кету керек. Әдепкі қалпы бойынша, gтер утилитасы метатаңбалардың кеңейтілген жинағын қолданбайды. gтер-те { } қолдану үшін сізге кеңейтілген gтер-ті қолдануға тура келеді. Бұл egтер немесе gтер-E. Біз gтер утилитасында толығырақ көреміз.

Осы сәтке дейін көрген барлық идеяларды біріктірейік және әлеуметтік сақтандыру нөміріне сәйкес келетін тұрақты өрнек жазайық. Әлеуметтік қамтамасыз ету нөмірі ### - ## -#### түріне ие, мұнда әрбір # сан. Осындай нөмірге сәйкес келетін тұрақты өрнек төменде келтірілген:

[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]

Бұл тұрақты өрнек саннан, саннан, саннан, дефистен, саннан, саннан, дефистен, саннан, саннан, саннан және саннан тұрады. {n}-ді қолдана отырып, біз өрнекті қысқарта аламыз:

[0-9]{3}-[0-9]{2}-[0-9]{4}

Ұялы телефон нөмірі қалай жазылады? Бұл біздің өрнектің қала кодымен немесе кодсыз нөмірге сәйкес келуін немесе қала кодымен және кодсыз телефон нөмірімен сәйкес келуін білгіміз келетіндігімізге тәуелді. Біз әзірше бұл сұраққа жауап бермейміз және оған тараудың соңына таман қайта ораламыз.

Басқаны қарастырып көрейік. IP-мекенжайын қалай сәйкестендіре аламыз? IP-мекенжайдың формасы 0-255.0-255.0-255.0-255. Келесі тұрақты өрнек дұрыс болып табылмайды. Неліктен екендігін анықтай аласыздар ма?

[0-255].[0-255].[0-255].[0-255]

IP-мекенжайдың тұрақты өрнегінің екі кемшілігі бар; біріншісі айқын, ал екіншісі аздап түсініксіздеу. [0-255] Нені білдіреді? Тұрақты өрнекте нұсқалар тізімін көрсету үшін []-ны пайдаланасыз. Нұсқалар ретінде нөмірленген тізім, мысалы, [abc] немесе [a-c] секілді ауқым бола алады. Бұл тұрақты өрнектің квадраттық жақшаларында ауқым да, нөмірленген тізім де қамтылған. Біріншіден, 0,1 немесе 2 сәйкес келетін 0-2 ауқымы бар. Екіншіден, 5,5 нөмірленген тізімі бар. Осылайша, жақшаға алынған әрбір элемент 0, 1, 2, 5 немесе 5-тің кез келгеніне сәйкес келетін болады. Осылайша, бұл тұрақты өрнек 0.1.2.5 немесе 5.5.5.5 немесе 0.5.1.2 сәйкес келетін болады. Ол 0.1.2.3-тен (жақшада 3 жоқ), 10.11.12.13-ке

(барлық жақшалар санды 13 секілді көп әріптік мән еместігін көрсетеді) сәйкес келмейтін болады.

Дұрыс нөмірленген тізімді қалай көрсетеміз? 0-ден 255-ке дейінгі барлық нөмірлерді айтып шығуға болады ма? Бұл оңай емес және тізімнің 256 саннан тұрғанын қаламас едік! Мынаған қалай қарайсыздар: [0-9]{1,3}. Бұл өрнек 0-ден 9-ға дейінгі кез келген жеке санға, 00-ден 99-ға дейінгі кез келген екі сандарға, сондай-ақ, 000-ден 999-ға дейінгі кез келген үш мәнді санға сәйкес келе алады. Өкінішке орай, бізде 255 және одан аз үш мәнді сандарды шектеуге арналған қарапайым тәсіл жоқ, себебі бұл 299.388.477.566 жолына сәйкес келетін болады. Бірақ қазір біз осы белгілеу функциясын қолданатын боламыз. Демек, өрнекті қайта жазайық:

[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}

Біздің айрықша тұрақты өрнегіміздің екінші кемшілігі . (кезеңді) қолдану болып табылады. "Кез келген бір таңбаға сәйкес келу" дегенді білдіретіндігін еске салайық. Егер біз кезеңді біздің тұрақты өрнегімізге қолдансақ, онда ол кез келгенге сәйкес келе алады. Осылайша, біздің жаңа тұрақты өрнегіміз 1.2.3.4 немесе 10.20.30.40 немесе 100.101.201.225-ке сәйкес келе алады, бірақ ол 1a2b3c4 немесе 1122334 немесе 1-2-3-4-ке, сондай-ақ IP-мекенжай болып табылмайтын көптеген басқа ретгіліктерге сәйкес келетін болады. Біздің мәселеміз біздің кезеңді метатаңба ретінде қабылдануын қаламайтындығымызда көрінеді; біз кезеңнің дәлме-дәл қабылдануын қалаймыз.

Әріптік таңбаны қалай көрсетеміз? Көптеген таңбаларды оларды дәлме-дәл қабылдау үшін атап шығуымыз керек. Мысалы, abc дәлме-дәл "abc" жолы болып саналады. Бірақ, егер таңбаның өзі метатаңба болып табылса, онда біз ерекше бірдене жасауымыз қажет. Екі жолы бар, біріншісі – оны []-ға орналастыру. Бұл []-ға бір таңбаны орналастыру үшін қолданбағанымен, [0-9]{1,3} [.] [0-9]{1,3} [.] [0-9]{1,3} [.] [0-9]{1,3} жарай береді. Метатаңба болып саналатын таңбаны көрсеткіміз келсе, оның мәнінен «кетуіміз» қажет. Бұл \ алдыңғы таңбасымен жасалынады: \. немесе \+ немесе \{немесе \\$. Осылайша, біздің соңғы жауабымыз:

[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}

Біз 255-тен жоғары үш мәнді сандарды қолдануға мүмкіндік бермейтін кемшілікті жоюға тырысайық.

Шығу таңбаларын қолданудың басқа да түрлері бар, олар басқарылатын жүйелік деп аталады. 10.3 кестесінде жалпы басқарылатын жүйелік тізімі келтірілген. Мысалы, егер сіз төрт жүйелі бос орынды кеңістіктерді тапқыңыз келсе, сіз \s{4}-ны қолдана аласыз. Немесе егер сіз сан-

сыз 1-ден 10-ға дейінгі кез келген жүйелілікке сәйкестікті тапқыңыз келсе, $\backslash D\{1,10\}$ -ны көрсете аласыз.

[]-ны басқаша қолдануға болады, бірақ оларды дұрыс қолдану қиынырақ. Егер сіз []-дағы нөмірленген тізімнің алдына ^ қойсаңыз, тізім енді "сәйкес келеді, егер осы таңбалардың бірде біреуі болмаса" деп түсіндіріледі. Сіз ешқандай сандар жоқ жолды табу үшін $[\wedge 0-9]^*$ -ды немесе жоғарғы регистрдағы әріптері жоқ жолды табу үшін $[\wedge A-Z]^*$ -ты қолдана аласыз. $[A-Z][a-z]+[\wedge A-Z]^*$ өрнегі ешқандай жоғарғы регистрдің қосымша әріптерінсіз бас әріпке, төменгі регистрдің кейбір әріптеріне ие болуы тиіс жолды табады. Бұл егер біз адам атын күтетін болсақ, орындалуы мүмкін, мысалы, Zappa деп жазылынуы мүмкін.

Неліктен $[\wedge \dots]$ -ны қолдану қиын? Мұны түсіндіру үшін, біз алдымен тұрақты өрнектердің немен сәйкес келетіндігін қайта қарап шығуымыз керек. Естеріңізде болсын, тұрақты өрнек – бұл басқа жолмен салыстыру үшін қолданылатын жол. Негізінде, тұрақты өрнектің сәйкес келетіні бұл – үлкен жолдың қосалқы жолы. Қосалқы жол бұл бар болғаны жолдың бір бөлігі ғана. Мысалы, егер, “Frank Zappa” жолы болса, келесі әрекеттердің бірі қосалқы жол болып саналады: “Frank”, “ank”, “Zappa”, “kZ”, “ppa”, “Frank Zappa”, және де “” (бос жол).

10.3 КЕСТЕ. Жалпы басқарылатын жүйеліліктер

$\backslash d$	Кез келген санға сәйкес келеді
$\backslash D$	Кез келген сан емеске сәйкес келеді
$\backslash s$	Кез келген бос орынды таңбаға сәйкес келеді
$\backslash S$	Кез келген бос орынды емес таңбаға сәйкес келеді
$\backslash w$	Кез келген әріпке немесе санға сәйкес келеді
$\backslash W$	Кез келген әріп емеске, сан емеске сәйкес келеді
$\backslash b$	Кез келген сөздің шегіне сәйкес келеді
$\backslash B$	Кез келген сөздің шегі емеске сәйкес келеді

$0\{1,2\}[a-zA-Z0-9]^+$ өрнегін қарастырайық. Бұл тұрақты өрнек әріптер мен сандардың кез келген комбинациясы болатын, бір немесе екі 0s-тен тұратын кез келген жолға сәйкес келеді.

0000abcd0000

Тұрақты өрнектерді алдында анықтағандықтан, бұл жол өрнекке сәйкес келмеуі тиіс, себебі, ол "бір немесе екі 0s"-ке ие емес, ол дәлме-дәл мән-де төрт 0s-ге ие. Дегенмен, өрнек тек соңында әріптер мен сандар бар "бір немесе екі 0s" ие, сәйкес келетін кез келген қосалқы жолды ғана іздейді. Жол "0a"-ны қамтитындықтан, ол сәйкес келеді. Тұрақты өрнек жолдағы әрбір таңбаға сәйкес келмеуі тиіс; ол тек тиісті қосалқы жолды табуы керек.

$[\wedge \dots]$ -ны қолдануға қайта орала отырып, мысалды қарастырайық. $[\wedge A-$

Z]+ өрнегін қарастырайық. Бұл өрнектің мәні түсінікті көрінеді: бірдеңеге сәйкес келеді, бас әріптер жоқ. Енді abCDefg жолын қарастырайық. Тұрақты өрнек шынымен де "бас әріптер сәйкес келмейді" деп тұрғандай, бірақ жол сондай-ақ кішкентай әріптерді де қамтиды. Осылайша, тұрақты өрнек жолдағы ab мен efg-ге сәйкес келетін болады, сондықтан жол сәйкес келетін ретінде табылды. Сосын егер [^AZ]+ бас әріптерді қамтитын жолға сәйкес келетін болса, оны қалай қолдануға болады? Осы тұрақты өрнек сәйкес келмейтін жалғыз жол – бұл тек жоғары регистр әріптерінен ғана тұратын кез келген жол. Ол abCDefg-ге сәйкес келгенімен, ол ABCDEFG-ге сәйкес келмейтін болады. [^ ...]-ны толық қолдану үшін өте абай болу қажет.

Осыны ескере отырып, 0 + 1 + өрнегін қарастырамыз. Ол 0001, 01111, 01-ге сәйкес келеді, бірақ ол 0101 мен 000111aaa-ге де сәйкес келетін болады, себебі, осы екі жолдар 0 + 1 + сәйкес келетін жүйелілікті қамтиды. Өрнек бойынша дәл сәйкестікті қалай қамтамасыз ете аламыз? Ол үшін бізге екі ^ және \$ қосымша метатаңбасы қажет. ^ жоғарыда көрсетілгендей, [] ішінде қолданыла алады және "сәйкес келеді, егер бұлар табылмаса" дегенді білдіреді. Бірақ []-дан тыс ^ "жолдың басында сәйкес келеді" дегенді білдіреді және \$ "жолдың соңында сәйкес келеді" дегенді білдіреді. Егер біздің тұрақты өрнегіміз ^өрнек \$ түріне ие болса, онда бұл сәйкес жолдың тек осындай форматқа ие бола алатындығын білдіреді.

Сандардан басталатын кез келген жолдардың сәйкес келгенін қалауымыз мүмкін. Бұған ^ [0-9] + тұрақты өрнегі арқылы қол жеткізуге болады. Біз * орнына + қолданамыз, себебі * "олардың ешқайсысына сәйкес келмейді" дегенді білдіреді, сондықтан ол саннан басталмайтын жолға сәйкес келетін болады. Біз штаттар аббревиатурасына аяқталатын кез келген жолдардың сәйкес келгенін қалауымыз мүмкін. Барлық мемлекеттік қысқартулар – бұл барлық бас әріптер, сондықтан [[A-Z][A-Z]\$ немесе [A-Z]{2} ретінде көрінуі мүмкін. Егер біз кезең түрінде аяқталатын аббревиатураны қалайтын болсақ, біз [A-Z] [A-Z] \. \$ қолдана аламыз және егер біз кезеңді қалауымызша жасағымыз келсе, біз [A-Z] [A-Z] \.? \$ қолданар едік. [A-Z][A-Z] көмегімен жоғарғы регистрдегі кез келген екі әрпінде сәйкестік таба аламыз, мысалы, AB мен ZZ заңды мемлекеттік қысқартулар болып табылмайды.

Жалпы, ^ және \$ метатаңбасының екеуін бір өрнекте қолданғымыз келмейді, себебі, бұл өте шектеуші болар еді. Осы уақытқа дейін қарастыратын түсініктерді көрсету үшін қызметкерлер туралы ақпаратты қамтитын файлды қарастырайық. Ақпарат жол-жол сайын берілген, әрбір қызметкердің тегі, аты, лауазымы, қабылданған жылы, кеңсе нөмірі

мен үй мекенжайы.

Нақты жылы, мысалы 2010 жылы қабылданған барлық қызметкерлерді табу үшін тұрақты өрнекті жазамыз. Біздің тұрақты өрнегіміз қарапайым 2010 түріне ие бола алады. Алайда, тек 2010-ды қолдану қате сәйкестіктерге әкелуі мүмкін, себебі, 2010 жолы адамның кеңсе нөмірі немесе мекенжай бөлігі ретінде пайда болуы мүмкін (мекенжай, пошта индексі).

Егер біз 2000 жылдан бастап қабылданған қызметкерлерді тапқымыз келсе не істеуге болады? Ол үшін тұрақты өрнек 20[01] [0-9] болуы мүмкін. Бұл қызметтік нөмір немесе мекенжайдың бөліктеріне сәйкес келуі мүмкін. Егер біз ^20 [01] [0-9]\$ қолданатын болсақ, біз 2000 мен 2019 арасындағы төрт саннан ғана тұратын жолдардың сәйкестігін шектейміз. Ешқандай жол сәйкес келмейтін болады, себебі, бір де бірі төрт мәнді нөмірге ие емес. Біз одан да ақылдырақ жасауымыз керек.

Жыл тегінен, атынан және лауазымынан кейін келетіндігіне назар аударыңыз. Біз қызметкердің тегін [A-Z] [A-Z] + ретінде көрсете алар едік. Біз атын [A-Z] [A-Z] * ретінде көрсете алар едік (біз 'J' ретінде жазылуға жол береміз). Біз лауазымды [A-Za-Z] + яғни кез келген әріптердің комбинациясы ретінде көрсете аламыз. Ақырында, біз осы жағдайда 20[01] [0-9] болуы керек қабылдау жылын алуды күтеміз. Егер жұмыскер туралы ақпараттың әрбір бөлігі үтір арқылы жазылған болса, онда біз біздің өрнегіміздің әрбір бөлігін үтірмен аяқталатындай етіп тұрғыза аламыз. 10.1 суретті қараңыз. Осында көрсетілгендей, қабылдау жылы болашақтағы жылдарды да қамтиды (2019 жылға дейін).

Біз осы іздеу мәселесін қатардың екінші шетінен шеше аламыз. Қабылдау жылы кеңсе нөмірі мен мекенжайдың алдында келеді. Егер біз кеңсе нөмірі тек сандардан, мекенжай тек мекенжайдан, қаладан, штаттан және пошта индексінен тұрады деп болжасақ, сосын жолды аяқтау үшін біз 20[01][0-9],[0-9]+ пен мекенжайды көруді күтеміз. Мекенжай неғұрлым күрделірек, себебі, ол әр түрлі компоненттерден, көше мекенжайынан, қаладан, аббревиатурадан (екі әріптен) және пошта индексінен тұрады. Көшенің мекенжайы әріптер мен бос орындар бар сандардың құрылымына ие болуы мүмкін (мысалы, 901 Pine Street). Сонымен қатар, мекенжайда пайда болатын кезеңдер болуы мүмкін (мысалы, 4315 E, Magnolia Road) немесе ол өзіне көшенің атауындағы сандарды қамтуы мүмкін (мысалы, 50 N. 10th Street). Ол сонымен қатар # таңбасын қолданатын пәтер нөміріне ие болуы мүмкін (мысалы, 242 Olive Blvd, # 6). Біз бәрін [A-Za-z0-9.#]+ секілді әрбір әлеуетті әріпті, санды, таңбаны және бос орынды аталған тізімдердің біріне орналастыру жолымен шешуге тырысып көрер едік. Қаланың атауы әріптерден ғана

тұруы тиіс (бос орындар мен кезеңдерді де қамтуы мүмкін), бірақ біз тек әріптерді ғана назарға алатын боламыз. 10.2 суреті қабылдау жылынан, тым болмағанда, 2000 басталатын, тек бес мәнді пошта индексі бар жолды аяқтау үшін қолдануға болатын толық тұрақты өрнекті көрсетеді (біз тоғыз мәнді пошта индексін қалай қолдануды төменде көреміз).

Соңғы метатаңбалар - | және (). Осы метатаңбаларды қолдану қандай да бір жүйелілікке сәйкес келуі мүмкін таңбалардың жүйелілігін қамтамасыз етеді. Яғни, біз заттар тізімін айтып шыға аламыз. []-дегі нөмірленген тізімге қарағанда, біз жүйелілікті тізіп шығамыз. Біз осы үш 2 әріпті қысқартулардың кез келгеніне сәйкестікті тапқымыз келеді: IN, KY, OH. Егер біз [IKO] [NYH] қолданғанымызда, онда ол IN, KY және OH-ге ғана емес, IY, IH, KN, KH, ON және OY-ке сәйкес келер еді. () ішіндегі жүйелілікті тізіп шығайық және нұсқалар арасына | қояйық. Осылай, (IN | KY | OH) дәлме-дәл мәнде үш жүйеліліктің біреуіне ғана сәйкес келетін болады.

() мен | қолдана отырып, біз бес мәнді де, тоғыз мәнді де пошта индексіне сәйкес келетін өрнекті қамтамасыз ете аламыз. Бес мәнді өрнек - [0-9] {5}. Тоғыз мәнді өрнек - [0-9] {5} -[0-9] {4}. Біз | көмегімен үйлестіреміз және оларды ()-ға орналастырамыз. Бұл бізге ([0-9] {5} | [0-9] {5} - [0-9] {4}) береді.

Біз IP-мекенжайдың мәселесін шешу үшін осылайша () мен | қолдана аламыз. Еске түсірейік, [0-9] {1,3} қолдана отырып, 255 көп емес үш мәнді санға сәйкес келетін болады, осылайша IP-мекенжайда жоқ сандарға сәйкес келетін болады. Біз (0 | 1 | 2 | 3 | ... | 254 | 255) секілді 256 мүмкін мәндердің бәрін тізіп шыққымыз келмейді, бірақ біз () мен | -ны басқа тәсілмен қолдана аламыз. [0-9] сандардың кез келгеніне сәйкес келеді деп ойланыңыз және IP-мекенжайлар кез келген бір мәнді санды қамтуы мүмкін. Біз [0-9] [0-9]-ны да қолданар едік, себебі, кез келген 00-99 жүйелілікті заңды IP-мекенжай болып табылады, біз әдетте жетекші нөлдерді қолданбаймыз. Осылайша, біз оны [0-9] {1,2}-ге дейін жеңілдетер едік. Дегенмен, бұл өзіне 100-255 жүйелілігін қамтымайды. Біз 100-255 бірнеше әр түрлі мүмкіндіктермен көрсете алар едік:

1[0-9][0-9]—this covers 100–199

2[0-4][0-9]—this covers 200–249

25[0-5]—this covers 250–255

10.3 суреті осы нұсқаларды бір ұзын тұрақты өрнекке орналастырады (бос орындар оларды оқу үшін аздап қолайлырақ ету үшін "|" таңбаларының жан-жағына қойылады).

Осы бөлімді бірқатар мысалдармен аяқтайық. Біріншіден, біз сәйкестендіргіміз келетін бірқатар жолдарды сипаттаймыз және оларға сәйкес

келетін тұрақты өрнектерді ойластырамыз. Екіншіден, бізде бірқатар тұрақты өрнектер болады және біз олар неге сәйкес келетіндігін түсіндіруге тырысамыз. Бізде студент туралы ақпараттар тізімін қамтитын, келесі түрдегі мәтіндік файл бар деп болжайық.

Student ID (a 16 digit number), last name, first name, major, minor, address. Негізгі және қосымша мамандықтар үш әріптік кодқа ие (мысалы, CSC, CIT, BIS, MIN), барлығы бас әріптер. Қосымша мамандық қажет, бірақ ол өзіне "мамандық таңдалмады" нұсқауы үшін үш бос орынды қамтиды. Мекенжай – мекенжай, қала, штат, пошта индексі.

Информатикада (CSC) немесе компьютерлік ақпараттық технологияларда (CIT) мамандығы бар барлық студенттерді тапқымыз келеді. Айқын жауап (CSC | CIT) іздеуде көрінеді. Алайда, бұл негізгі және қосымша мамандықтар арасында ешқандай айырмашылық жасамайды, біз тек негізгі мамандықтарды тапқымыз келеді. Байқасаңыз, негізгі мамандықта 16 мәнді нөмір мен ат бар. Мысалы, біз 0123456789012345, Zappa, Frank, CSC, ... көруді күтуіміз мүмкін. Сонда біз жолдың басынан бастап тұрақты өрнекті жазар едік:

`^[0-9]{16}, [A-Z][a-z]+, [A-Z][a-z]+, (CSC|CIT)`

Біз оны қысқарта аламыз. Қосымша мамандықтар алдында әрқашан үш бас әріптен тұратын негізгі мамандық тұратындықтан, біз қосымша мамандықтың алдынан емес, негізгі мамандық алдынан көруді күтеміз, бұл толықтай капиталға айналмайды. Осылайша, біз өрнекті келесідей қысқарта аламыз:

`[A-Z][a-z]+, (CSC|CIT)`

Мұнда бізге бас әріптің келесі кішкентай әріптердің жүйелілігі (аты) қажет, сосын үтір арқылы CSC немесе CIT қойылады. Егер CSC немесе CIT қосымша мамандыққа сәйкес келсе, оның алдындағы жол өзіне кішкентай әріптерді қамтымайды, ал егер бас әріптерден кейін тегін немесе пошта мекенжайына сәйкес келетін кішкентай әріптер болса, олардан кейін CSC де, CIT да болмайды.

Пәтерлерде тұратын барлық студенттерді тапқымыз келеді. Біз apt, apt., # немесе apartment-ті қолдана отырып, пәтерлер мекенжай бөлімінде көрсетілген деп болжам жасаймыз. [.]? (немесе \.?) көмегімен біз apt-дан кейінгі кезең міндетті болып табылмайтындығын көрсете аламыз. біз оларды келесідей атап шыға аламыз:

`([Aa]pt[.]?|[Aa]partment|/)`

Біз KY немесе OH қысқартулары мекенжай жолында бір жерден пайда болады деп күтпедік. Дегенмен, KY не OH қысқартылған үш әріптің бір бөлігі ретінде қолдануға болады. Егер біз іздестіру жолдарын қауіпсіз түрде жүргізіміз келсе, онда қысқартылған сөздер жолды аяқтайтын

пошта индексінің алдынан бірден пайда болуы тиіс. Ол үшін біз келесіні көрсете аламыз:

(KY|OH), [0-9]{5}\$

немесе егер пошта индексі бес мәнді және кеңейту төрт мәнді болады деп болжасақ, келесіні қолданар едік:

(KY|OH), ([0-9]{5}|[0-9]{5}-[0-9]{4})\$

Қысқарту бос орын қалған жағдайда пайда болады және егер ол кезеңді аяқтаса, онда балама ретінде біз бос орын мен кезеңді қоса аламыз, сондықтан үш әріпті қысқарма сәйкес келмейді:

(KY\.|OH\.)

Егер негізгі ретінде SKY немесе OHM-ды алатын болсақ, бұл осы штаттардың аббревиатураларына кеңістік пен екі әріп қоршап тұратын кезеңге байланысты сәйкес келмейтін болады:

Спам-сүзгілер мен тұрақты өрнектер

Егер сіз электронды пошта арқылы қаламаған хабарламаларды сүзгілеу үшін спам-сүзгілер жасағыңыз келсе, бұл қарапайым нәрсе. Электронды пошта арқылы кейбір сөздерді іздейтін бағдарламаны (мәтіндік файлды) жазу қажет: "қолма-қол ақшалар", "әрекет ету мерзімі", "арығу", "виагра", "үйге жұмыс", "сіз таңдалдыңыз" және т.б. Бірақ кейбір спамдар негізгі сөздерді жасыра отырып, өтіп кетеді.

Электронды пошта бойынша спамды, жарнаманы, арзан әрі қолжетімді виагра дәрілерін қарау. Электронды пошта мекенжайын жасыруға тырысуға болады, мысалы, «Виагра» сөзін әр түрлі формалар арқылы жасыруға болады: V!agra, V.i.a.g.r.a, Vi@ gra, V!agra, ViaSEXYgra және т.б. Егер біз әріптік ауыстырулар туралы ғана ойласақ, онда біз біздің тұрақты өрнегімізде барлық мүмкін ауыстыруларды атап шығуға тырысуға болады, мысалы:

[Vv][i!][Aa@][gG9][Rr][aA@]

Сипаты бізге белгісіз ауыстыруларға қатысты, мысалы, 3-тің орнына 'e' немесе 1 немесе ! орнына - 'i'? Мысалы, V##gra немесе Viag^^? Мұнда біздің тұрақты өрнектерімізбен абай болу керек. Біз, мысалы, Vv].{2} gra, [Vv]iag.{2} және басқа да нұсқаларды қолдана аламыз, бірақ заңды сөздерді бұғаттамау үшін мұқият болу қажет. Мысалы, [Vv]ia.{3} спам бола алмайды.

V.i.a.g.r.a., ViaSEXYgra немесе Via##gra секілді қосымша таңбалар қой-

ылған осы сөздің нұсқасына қатысты не айтасыз? Осы мәселені шешу үшін, біз * белгілеуін әріптер арасына қоя аламыз. Естеріңізге салайық, кез келген таңба мен * 0-ді білдіреді. Осылайша, V.*i.*a.*g.*r.*a.*-ны олардың арасында пайда болатын таңбаларға тәуелсіз ‘V’, ‘i’, ‘a’, ‘g’, ‘r’, ‘a’ әріптеріндегі сәйкестікті іздеу қажет. * қолдануына байланысты, біз осы әріптер арасында ештеңе жоқ жолдарды салыстыра аламыз. Тұрақты өрнектерсіз, спам-сүзгілер табыстылығы неғұрлым азырақ болар еді. Бірақ, сіз көріп отырғандай, тұрақты өрнектерді анықтау сүзгіге арналған күрделі міндет болуы мүмкін!

Біз пошта индекстері 41-мен басталатын барлық студенттерді тапқымыз келеді. Ол өте оңай:

41[0-9]{3}

Индекс нөмірі (мысалы, 01234123456789012) ғана сәйкес келгенімен немесе іздеуші индексті мекенжаймен, көшемен немесе үймен шатастырып алуы мүмкін. Сондықтан кодты жолдың соңына ауыстырып, оны келесінің көмегімен түзету қажет:

41[0-9]{3}\$

Немесе төрт мәнді кеңейту көрсетуі мүмкін:

(41[0-9]{3}|41[0-9]{3}-[0-9]{4})\$

Біз ID жұп санмен аяқталатын кез келген студентті тапқымыз келеді. Біз [02468] қолдана алмаймыз, себебі, бұл кез келген осы сандарды қамтитын кез келген жердегі (сәйкестендіру нөміріндегі, мекенжайдағы, пошта индексіндегі) кез келген жолға сәйкес келетін болады. Бірақ студенттер саны әрқашан да 16 саннан тұрады, сондықтан біз 16 мәнді ID-ді ғана тапқымыз келеді:

[0-9]{15}[02468]

Біз тек ID іздейтіндігімізге көз жеткізу үшін (іздеуші бізге 16 және одан да көп саннан тұратын, мысалы, мекенжайды беруінің алдын алу үшін) басында ^ қояр едік.

Соңғы мысал. Біз тегі Заппа деген оқушыны тапқымыз келеді. Бұл оңай: Заппа

Заппа деген не? Бұл адамның атын немесе оның мекенжайын көрсетеді? Әрине. Екіталай, бірақ болуы мүмкін. Студенттің тегі әрқашан оның ID-нан кейін жазылады. Сондықтан, біз студенттің оның ID-мен сәйкестігіне көз жеткізе аламыз : ^[0-9]{16}, Заппа.

Осыны басқа жағынан қарастырайық. Міне біршама тұрақты шамалар. Осында сәйкестікте қателер болады ма?

([^O][^N], [0-9]{5}\$[^O][^N], [0-9]{5}-[0-9]{4}\$)

Біз жолдың соңындағы бес немесе тоғыз мәнді нөмірде жазылатын ОН

аббревиатурасына ие емес барлығын таптық. Осылайша, біз Огайода тұрмайтын студенттерді анықтадық.

[A-Z]{3},[]{4},

Бұл жол артынан үтір болатын, үтірден және төрт бос орыннан кейін (үтірден кейінгі бос орын) келетін үш бас әріпті табатын болады.

Біз негізгі немесе қосымша іздеу нәтижелері үшін үш бас әріпті ғана көруді күтеміз.

^0{15}[0-9]

Мұнда біз 15 нөлмен басталатын жолды іздейміз және одан кейін кез келген басқа сандар болады. Егер студенттің ID нөмірі дұрыс рәсімделсе, онда бастапқы 10 студентті сәйкестендіруге болады.

ПӘРМЕНДІК ИНТЕРПРЕТАТОР (ТҮСІНДІРУШІ) ЖӘНЕ АЛМАСТЫРЫЛАТЫН БЕЛГІЛЕР

9 тараудан Bash интерпретаторының нұсқаулықты орындау кезінде бірнеше кадамды орындауды еске түсірейік. Осы кадамдардың ішінде көптеген кеңейтулер бар. Олардың бірі – файл атын кеңейту. Егер пайдаланушы файл атында (немесе жол атын) топтық таңбаны көрсететін болса, Bash тиісті файлдар немесе каталогтар тізіміне алмастыру таңбасын өрістетеді. Сосын бұл тізім бағдарламамен бірге беріледі. Мысалы, Ls * .txt-тен, онда Ls пәрмені конвертациялайды орындамайды, бірақ оның орнына Bash * .txt-ты сәйкес келетін барлық файлдар тізіміне конвертациялайды, сосын Ls-ке барлық тізімді береді. Ls пәрмені Linux көптеген пәрмендері секілді бір элементте немесе тізімде бірдей жұмыс істеуі мүмкін.

Өкінішке орай, Bash-тағы алмастыру таңбаларын белгілеу үшін қолданылатын таңбалар метатаңбалардың тұрақты өрнектерінің бірқатары секілді. Бұл жаңылысқа әкелуі мүмкін, себебі, көптеген пайдаланушылар тұрақты өрнектерді білгенге дейін топтық таңбаларды зерттей бастайды. Сіз картаны бірінші болып зерттегендіктен, сіз Ls * .txt-тегі *-ны "бірдеңеге сәйкес келу" секілді түсінідіріледі.

Метатаңбалар туралы жоғарыда айтылғандай, * метатаңбасы "алдыңғы таңбаға нөл немесе бірден көп сәйкес келу" дегенді білдіреді. Ls * .txt жағдайында алдыңғы таңба бос кеңістікті білдіреді. Бұл дұрыс түсіндірмеу болар еді. Сондықтан, Linux пайдаланушысы немесе әкімші ретінде сіз Grep секілді бағдарламалардағы тұрақты өрнектер ретінде қолданылуына байланысты жаһандануды орындау үшін Bash-та қолданыла-

тын алмастыру таңбаларын қолдануды ажырату жағдайында болуыңыз қажет.

Bash-та қолданылатын алмастырылатын белгілер – *, ?, +, @, !, \, [], [^...] және [[...]]. Олардың кейбіреулері қарапайым өрнектер жинағына метатаңбамен қамтылған, ал кейбіреулері қамтылмаған. Біз осы бөлімде жалпы топтық таңбаларды қолдануды қарастырамыз. 10.4 кесте олардың әрқайсысы үшін түсіндірмені қамтамасыз етеді. Таңбалардың сәйкестік үлгілерінің кеңейтілген жинағын қолдану үшін, Bash орнатқан кезде ғана қолжетімді, кестеде белгіленген топтық таңбаларға назар аударыңыз. Біз болжап отырғандай, бұл сіздің Bash ортаңызда орнатылмады, біз олардың мәні түсінікті болуы қажет болғанымен, біз ол таңбаларды қарастырмаймыз.

Біз осы бөлімді 10.4 кестедегі алмастырылатын таңбалардың бірқатары қолданатын бірқатар мысалдармен аяқтаймыз. Біз кеңейтілген жинақтағы топтық таңбаларды алып тастаймыз. Осы мысал үшін бізде ағымдағы жұмыс бағытындағы келесі файлдар мен ішкі тізімдемелер бар деп болжаймыз. Кіші тізімдемелер олардың атымен/алдында көрсетілген.

10.4 кестесі. Bash-та қолданылатын метатаңбалардың мағыналары

*	Бос жолды қоса алғанда, кез келген жолға сәйкес келеді
**	Барлық файлдар мен тізімдемелерге сәйкес келеді
**/	Тізімдемелерге сәйкес келеді
?	Кез келген бір таңбаға сәйкес келеді (ескерту: 0 таңбаларға сәйкес келмейді)
+	Бір немесе көп кірулерге сәйкес келеді (оның тұрақты өрнектерде қолданғанына ұқсас)
@	Тізімдегі кез келген мысалға сәйкес келеді
!	Тізімдегі бір мысалдан басқа ештеңеге сәйкес келмейді
\	Тұрақты өрнектермен осы таңба мәнінің ұқсастығын болдырмау үшін қолданылады, мысалы, * *-мен сәйкес келмейді дегенді білдіреді
[...]	Жабық таңбалардың кез келгеніне сәйкес келеді, дефисті қолдану кезінде ауқымдарға рұқсат етіледі, егер бірінші таңбадан кейін ол жақшаға алынбаған кез келген таңбаға сәйкес келеді

Bash-тағы кеңейту жағдайындағыдай, Ls {C, H} *-дегідей "барлығын жинау" дегенді көрсету үшін тізімдер {}-ге {...} орналастырылуы мүмкін

Тұрақты өрнектер секілді берілген сыныптағы кез келген [[:class:]] таңбаға сәйкес келеді.

```
foo foo.txt foo1.txt foo2.dat foo11.txt /foo3 /fox /foreign /foxr FOO FOO.txt
FOO1.dat FOO11.txt foo5?.txt /FOO4
```

Әрбір мысалы бар 10.5 кестесін қараңыз. Кестеде Linux, Ls пәрмендері мен тізімдемеден элементтер келтірілген.

Грег бағдарламасы

Грег бағдарламасы берілген тұрақты өрнекке сәйкес келетін жолдарға арналған бір немесе бірнеше мәтіндік файлдарды іздейді. Ол осындай файлдар табылған жолдарды басып шығарады. Осылайша, пайдаланушы немесе әкімшілік қалаған құрылымға сәйкес келетін мәтіндік файлдардан жолдарды жылдам ала алады.

10.5 кесте Мысалдар

Ls пәрмендері	Элементтер
ls *.txt	foo.txt, foo1.txt, foo11.txt, FOO.txt, FOO11.txt, foo5?.txt
ls *.*	foo.txt, foo1.txt, foo2.dat, foo11.txt, FOO.txt, FOO1.dat, FOO11.txt, foo5?.txt
ls *	Тізімдемеден барлық тізімді атап шығады
ls foo?.*	foo1.txt, foo2.dat
ls foo??.*	foo11.txt, foo5?.txt
ls *\?.*	foo5?.txt
ls *. {dat,txt}	either.txt or .dat аяқталған тізімдемедегі барлық тізімді атап шығады
ls foo[0-2].*	foo1.txt, foo2.dat
ls*[[upper:]].txt	FOO11.txt, FOO.txt
ls*[[upper:]]*	FOO, FOO11.txt, FOO1.dat, FOO.txt/FOO4
ls *[[digit:]]*	Саны бар әрбір элементті атап шығады
lsfoo[[digit:]].*	foo1.txt, foo2.dat (foo11.txt атап шықпайды, себебі, біз тек 1 санды ғана іздейміз, foo5? атап шықпайды.

Метатанбалар туралы бөлім соңында біз файлдағы нақты студенттік жазбаларды анықтау үшін қолданылатын тұрақты өрнектерді қарастырдық. Grep секілді бірнеше файлдармен бір уақытта жұмыс істей алатындай, Grep әрбір салыстыру үшін екі затты, сәйкестікті қамтуы мүмкін файл аты мен жолды қайтарады. -n опциясының көмегімен сіз әрбір сәйкестік үшін жолдың нөмірін ала аласыз. Егер әрбір жол, файлдың бірнеше жолдары сәйкес келсе немесе ешқандай жолдар сәйкес келмесе, сіздің тұрақты өрнегіңіз бен файлдағы жолдарға байланысты Grep-ті қолданған кезде бағдарлама барлық файлды қайтара алады.

Grep/ бағдарламасы кеңейтуді келесідей береді:

grep pattern filename(s)

Егер сіз бірнеше файлдарды қолданғыңыз келсе, жоғарыда 2 бөлікте айтылғандай сіз * немесе ? қолдана аласыз немесе сіз бос орындармен бөлінген файлдардың бірнеше аттарын атап шыға аласыз. Егер сіздің үлгіңізде бос орындар болса, үлгіні " немесе "" қосу қажет. Бұл өзінің тұрақты өрнектерінде сақтық шарасы ретінде әрқашан " немесе "" қолдану жақсы әдет.

Негізінде, "-ны қолдану неғұрлым ұнамды болып табылады. Бұл Bash интерпретаторы GREP жасай алмайтындай бірнеше таңбалармен түсіндіреді. Grep !! файл аты тұжырымын қарастырамыз. Бұл тұжырым қарапайым көрінеді, кейіпкерлер үшін файл атауын іздеу !!. Өкінішке орай, !! таңбалары Bash интерпретаторына арналған және соңғы пәрмендер қайтарылып алынуы тиіс. Нұсқаулық cd ~ болды деп ойлаңыз. Себебі, Bash интерпретаторы пәрменді орындау алдында !! секілді осындай элементтерді қайтаратын болғандықтан, пәрмен grep cd ~ файл атына өзгереді. Осылайша, Grep cd ~ таңбаларының жүйелілігі үшін файл атын іздейтін болады.

Басқа мысал \$-ден пайда болады. Сіз \$ Bash-тағы айнымалылар атының алдында келетіндігін еске түсірсеңіз болады. !! жағдайындағыдай, Bash интерпретаторы пәрменді орындау алдында айнымалыларды олардың мәндерімен алмастырады. Осылайша, \$ HOME файл аты пәрмені Grep/ home/пайдаланушы аты файл аты пәрменіне ауыстырылады. Осы мәселелерді болдырмау үшін, Bash интерпретаторы кез келген ашылулар немесе интерпретацияны болдырмау үшін жалғыз жақшаларды шақырады. Осы мәселенің алдын алмай үшін екі рет айта отырып, "\$ HOME" бұрынғыдай \$ HOME-де сақталатын мәнге түрленеді.

Grep осы файлдың әрбір жолы үшін тұрақты өрнекті салыстыруға тырысады. Бұл, бәлкім, біз бастапқыда тұрақты өрнектерді қолдану туралы болжағандай емес, себебі біз сызықтар емес, жолдарға сәйкестік ретінде тұрақты өрнекті сипаттадық. Негізінде, Grep файлдың әрбір жолын жол

түрінде қарастырады және әрбір қосалқы жолдың тұрақты өрнегімен сәйкестікке салыстырады. Үлгідегі бірнәрсе осы жолда сәйкес келеді немесе сәйкес келмейді. Grep бағдарламасы файлдың әрбір жеке жолын іздемейтін болады (жолдар бос орындармен бөлінеді деп болжанады).

Мысалы, егер файл жолға ие болса:

bat bait beat beet bet bit bite boot bout but

Біз b.t кем дегенде бір жолға сәйкес келгеннен бастап b.t тұрақты өрнекті қолдандық, Grep барлық жолды қайтарады.

Grep бағдарламасы тұрақты метатаңбаларды қолданады. Ол өзіне {}, | қамтитын кеңейтілген тұрақты өрнектер жинақтарын қамтымайды. Дегенмен, Grep опцияны, ГРЭП -Е (немесе бағдарламаны) қамтиды, кеңейтілген жинақты қолданатын кеңейтуді береді.

Grep| бағдарламасы кеңейтуді қалай беру керек:

tokenizer filename | grep b.t

Мұндай жағдайда Grep файл(дар) атын алмайды, бірақ орнына файл туралы ақпарат тікелей оған келіп түседі.

Таңба мәнінен құтылу үшін \ қолдану қажеттілігі басқа жағдай. Біз мұны таңба метатаңба болып табылатын жолдағы таңбаларды дәлме-дәл мәнінде іздеу қажет болғанда қарастырған едік (метатаңбалар).

Мысалы, \$ "жол соңы" дегенді білдіреді, бірақ егер біз \$ іздейтін болсақ, біз мұны \ \$ ретінде көрсете аламыз. Бұл Tings басқа жинақтарындағы тұрақты өрнектерді беру кезіндегідей Grep-те дұрыс. Дегенмен, Grep-те ерекшелік бар. Мысалы, егер дефис элементтер тізімінде біреуі ретінде сұратылса, бұл келесі түрдей көрінуі мүмкін: [@ # % & - = < > !].

[] ішіндегі дефис ауқымды белгілеу үшін қолданылатындығын есте сақтаңыздар; сондықтан оның орнына біз бұл тізімді [@ # % & \ - ! = < >] ретінде көрсете аламыз. \ - ауқым ретінде қолдану үшін емес, дефистің дәлме-дәл мәнінде ізделетіндігін көрсету үшін қолданылады. Бірақ осы ережеден бір ерекшелік бар. Егер дефис жолдың соңында пайда болса, \ қолдану қажет етілмейді, сондықтан тізім [@ # % & = < > - !] болуы мүмкін.

Қалқалау таңбасы (\) қажет болмайтын жағдайлар да бар. Олардың біреуі \$-ға қамтылатын таңбалар тізім болып табылады, бірақ \$ таңбалар тізімінің соңында болмайды. Егер біз "жол соңын" көрсету үшін \$ қолданатын болсақ, Grep одан кейін қандай да бір белгілерді күтпейді. Сондықтан, егер біз \$ [0-9] + (соманы долларда көрсету) ие болсақ, Grep \$-ды "жол сәйкестігінің соңы" ретінде емес, доллардың шынайы белгісі ретінде қабылданады. Осы ^-ға да қатысты. Ақырында, егер метатаңба [] ішінде болса, көптеген таңбалар мәнін жоғалтады, сондықтан, мысалы, \$ немесе ? іздейтін болсақ, [\ \$] немесе [?] енгізбеуіміз қажет; оның

орнына біз [\$] немесе [?] қолдана аламыз.

Қазіргі кездегі Grep-ті (кеңейтуді беру) қарастырайық. 10.4 суретте [0-9] {1,3} \. [0-9] {1,3} \. [0-9] {1,3} \. [0-9] {1,3} / және т.б. / * кеңейтуін беру -п пәрменін орындау нәтижелерінің бір бөлігі көрсетілген. Тұрақты өрнектер нүктелермен бөлінген төрт санды алу үшін алдында жасаған өрнектер, яғни IP-мекенжайды алуға арналған төрт октетіміз. Бұл тұрақты өрнек IP-мекенжай болып табылмайтын (мысалы, 999.999.999.999) элементтерді қайтаруы мүмкін, бірақ мұндай жолдар / және әрі қарай тізімдемеде табылмайды. 10.4 суреті барлық нәтижені көрсетпейді, себебі Grep пәрмені өте көп сәйкестіктерді қайтарады.

Шығар кезде біз әрбір сәйкестік үшін файл атын, сәйкестік жолының нөмірін және жолдың өзін көретіндігімізге назар аударыңыз. Сіз / және әрі қарай файлын байқай аласыз. / Muttrc шындығында IP-мекенжайды қамтымайды; оның орнына, бағдарламалық қамтамасыз ету нұсқаларының нөмірлерін (-1.4.2.2) қамтиды. Тіпті егер біз біздің "дұрыс" тұрақты өрнектерімізді IP-мекенжайларының сәйкестігі үшін қолдансақ та (10.3 сурет), біз әліге дейін Muttrc файлына сәйкеспіз.

```
/etc/Muttrc:macro pager <f1> "!less /usr/share/doc/mutt-1.4.2.2/manual.txt\n" "Show Mutt documentation"
/etc/Muttrc:macro generic <f2> "!less /usr/share/doc/mutt-1.4.2.2/manual.txt\n" "Show Mutt documentation"
/etc/Muttrc:macro index <f2> "!less /usr/share/doc/mutt-1.4.2.2/manual.txt\n" "Show Mutt documentation"
/etc/Muttrc:macro pager <f2> "!less /usr/share/doc/mutt-1.4.2.2/manual.txt\n" "Show Mutt documentation"
/etc/ntp.conf:restrict 127.0.0.1
/etc/ntp.conf:#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap
/etc/ntp.conf:#broadcast 192.168.1.255 key 42 # broadcast server
/etc/ntp.conf:#broadcast 224.0.1.1 key 42 # multicast server
/etc/ntp.conf:#multicastclient 224.0.1.1 # multicast client
/etc/ntp.conf:#manycastserver 239.255.254.254 # manycast server
/etc/ntp.conf:#manycastclient 239.255.254.254 key 42 # manycast client
/etc/ntp.conf:server 127.127.1.0 # local clock
/etc/ntp.conf:fudge 127.127.1.0 stratum 10
Binary file /etc/prelink.cache matches
/etc/resolv.conf:nameserver 172.28.102.11
/etc/resolv.conf:nameserver 172.28.102.13
/etc/resolv.conf:nameserver 10.11.0.51
/etc/resolv.conf:nameserver 10.14.1.10
/etc/resolv.conf.predhclient:nameserver 172.28.102.11
/etc/resolv.conf.predhclient:nameserver 172.28.102.13
/etc/resolv.conf.predhclient:nameserver 10.11.0.51
/etc/resolv.conf.predhclient:nameserver 10.14.1.10
/etc/termcap:# GNOME Terminal 1.4.0.4 (Redhat 7.2)
```

10.4 сурет. Grep-тен нәтиже.

Біз IP-мекенжайдың тұрақты өрнегінің алдына бос орынды қоса отырып, бұдан құтылуға болады. 10.4 суретте нәтижелері көрсетілген пәрмен түбір ретінде көрсетілуі керек. Бұл файлдардың көпшілігі және әрі

қарай / соңғы пайдаланушымен оқылмайды. Егер соңғы пайдаланушы Grep-ке пәрмен берсе, сол элементтердің көпшілігі қайтарылар еді, бірақ кателер туралы келесі хабарламалар да қайтарылар еді:

egrep: security: Permission denied egrep: shadow: Permission denied egrep: shadow-: Permission denied egrep: sudoers: Permission denied egrep: tcsd.conf: Permission denied

Сонымен қатар, /etc/resolv.conf алдында ерекше жолды байқауға болады. Ол "екілік файл /etc/prelink.cache matches" оқиды. Бұл бізді екілік файлдың ұқсастықтарды қамтығандығы туралы ақпараттандырады. Дегенмен, біз қандай да бір екілік шығуды көргіміз келмейді, бізге осы файлдың ішіндегі нақты ұқсастықтар көрсетілмеген. Біз Grep-ті бинарлы файлдардан ақпарат шығаруға мәжбүрлей аламыз (10.6 кесте). Grep бағдарламасының бірқатар пайдалы опциялары бар; олардың кең таралғандары 10.6 кестеде келтірілген. -c, -E, -i, -n және -v атап өткен жөн. Кейінірек -v туралы талқылаймыз.

Бірқатар мысалдарды қарастырайық. Біз әр түрлі ұқсастықтарды іздеу үшін кеңсе факультетінің ақпарат файлын қолданатын боламыз. Бұл файл, offices.txt, әрбір факультет үшін ақпаратты, олардың кеңселерінің орналасуын (ғимарат аббревиатурасы мен кеңсе нөмірін), олардың тегін, сосын олар қолданатын компьютер(лер) платформасын сақтайды.

-a	Екілік файлда мәтіндік файл болса, оны өңдеу (бұл екілік сандардың нақты жолдары үшін бинарлы файлдарды іздеуді жүзеге асыруға мүмкіндік береді)
-c	Сәйкестіктер санын есептеу және жалпы санын шығару
-d read	Осы тізімдеменің барлық файлдарын өңдеу үшін қолданылады, осы тізімдеменің және барлық ішкі тізімдемелер үшін рекурсивті барлық файлдарын оқып шығу үшін оқудың орнына рекурсияны қолданыңыз
-E	Кеңейтуді қолдану (бірқатар тұрақты өрнектерді кеңейтуге көмектеседі)
-e regex	Тұрақты өрнек әдетте нұсқаулықта орналасатын жерде емес, -e-ден кейін орналасады; бұл тұрақты өрнек әдеттен тыс басталса, мысалы, дефиспен басталғанда оны қорғау үшін қолданылады
-h	Шығар кезде файл атын қысу
-i	Регистрді елемей (мысалы,[a-z] жоғарғы немесе төменгі регистрдегі кез келген әріпке сәйкес келеді)
-L	Сәйкестіктері жоқ файлдарды ғана шығару, сәйкестіктерді шығармау
-m	
NUM	NUM сәйкестіктен кейін файлды оқуды тоқтату
-n	Жолдың нөмірін шығару

-o Тұрақты өрнектермен сәйкестіктері бар жолдардың бөлігін ғана шығару

-R, -r Рекурсивті іздеу

-v Сәйкестікті айналдыру

Әрбір элемент үтірмен бөлінеді және компьютер платформасы бірнеше элементтерден тұруы мүмкін. Платформа үшін параметрлер - PC, Mac, Linux, Unix. Кеңселер мекенжайлары екі немесе үш әріптен тұратын шартты белгіге ие, мысалы ST, MEP немесе GH, сосын бос орын. Өзінің ғимаратының үшінші қабатында барлық жазбаларды табу үшін Grep пәрменін жазу (кеңсенің үш мәнді нөмірі 3xx болады деген шартпен).

```
egrep '3[0-9][0-9]' offices.txt
```

PC компьютерлері бар оқытушылардың барлық жазбаларын табатын Grep пәрменін жазу.

```
egrep 'PC' offices.txt
```

Өзінің үш әріптік аббревиатура шегінде ДК-мен тапсырма болса, не істеу қажет?

Бұл сәйкес келетін болады. Біз оның орнына мынаны қолданып көрер едік:

```
egrep 'PC$' offices.txt
```

Бұл өрнек PC жолдың соңында болған жолдарға ғана сәйкес келетін болады. Дегенмен, егер оқытушыда бірнеше компьютерлер болса және ДК тізімнің соңында болмаса, осы адамды ұстаудың қажеті болмайды. Жол келесідей болатындығын ескеріңіз: 123 ABC, Fox, PC, Mac. Егер біз ол ДК-ға сәйкес келгенін қаласақ, ол үтір мен бос орыннан кейін сәйкес келуі тиіс. Бұл тапсырманың келісілуінен құтылуға мүмкіндік береді, мысалы 456 немесе PCA 789 APC. Осылайша, біздің Grep жаңа пәрменіміз – бұл:

```
egrep ', PC' offices.txt
```

Егер факультеттің мүшесінде бірнеше компьютер болса, олардың әрқайсысы компьютер, компьютер, компьютер және әрі қарай ретінде аталып шықса. Сондықтан, егер бізде компьютер тізімнің кез келген жерінде болса, бұл үтір мен бос орыннан кейін орындалады. Бірақ егер PC ғимаратта пайда болса, ол сандар мен бос орын (456 PCA секілді) немесе жоғарғы регистрдегі әріптер (789 APC) кейін пайда болады. Бірден көп компьютері бар оқытушылардың барлық жазбаларын табатын Grep пәрмені. Кеңейтілген іздеу пәрменіндегі "есептеуіш" амалын қосудың тәсілі бар ма? Белгілі бір мән жағынан алғанда, бар. {п, т} қолдану кезінде күтілетін сәйкестіктердің санын бақылай алатындығымызды еске са-

лайық. Бірақ шынымен де неге сенгіміз келеді? Жолдың екі мысалын қарастырайық: бір компьютері бар факультет және бірден көп компьютері бар факультет.

123 MEP, Newman, Mac

444 GH, Fox, PC, Linux

Бір компьютерден, бірінші жазбада тек екі үтір бар. Екі компьютердің көмегімен, екінші элементте үш үтір бар. Бұл бізге тым болмағанда үш үтірі бар жолдарды табу қажеттігі туралы айтады. Дегенмен, '{3}' адекватты тұрақты өрнек болып табылмайды, себебі, ол тым болмағанда үш жүйелі үтірі бар (мысалы, 456 PCA, Заппа, PC, Mac) жолға ғана сәйкес келетін болады. Біз әрбір үтірдің алдына қойылу үшін кез келген таңбаларға рұқсат беруіміз керек. Шындығында, жолдарға қатысты біздің мысалға қарап, біз үтір алдында бос орын мен әріптер (сандар емес, себебі сандар ғана жолдың басында келеді) бар екендігін ойлаймыз. Біздің тұрақты өрнегіміз, яғни:

egrep '([A-Za-z]+,){3,}'

Біздің ()-ны қолдануымызға назар аударыңыз. Бұл {3} барлық тұрақты түрлендірулерде (үтір болып табылатын алдыңғы таңбада ғана емес) қолданылуы үшін қажет. Біз шындығында метатаңбаны қолдана отырып, мұны жеңілдетер едік. . кез келген бір таңбаға сәйкес келе алатындығын еске салайық. Сипаттың кез келген түрінен бір немесе бірнешеуін анықтау үшін + қоса аламыз. Біз барлығымыз тым болмағанда үш үтірді табуға мүдделі болғандықтан, біз '. +. +. + ' қолдана аламыз немесе (. +) {3} қолданар едік. Linux немесе Unix машинасы бар оқытушылардың барлық жазбаларын табатын Grep пәрменін жазу. Мұнда біз (Linux | Unix)-дағы секілді опцияны қолдана аламыз. Біз сонымен қатар осы сиқырды [] нұсқаларының көмегімен тырысып көруге болады. Бұл келесідей болады: [L] [IU] п [Ui] x. Екі GREP пәрмені болып табылады:

egrep '(Linux|Unix)' offices.txt egrep '[L]?[iU]n[ui]x' offices.txt

Кеңсесінде ДК жоқ оқытушылардың барлық жазбаларын табатын Grep пәрменін жазу. Айқын шешім өрнекте [^ P] [^ C] қолдану болар еді. Яғни, P-да келесі C жоқ жолды табу. Өкінішке орай, бұл өрнек файлдың

әрбір кеңейтілген жолын қайтарады. Неге? Себебі, тұрақты өрнек компьютері жоқ кез келген жолды береді. Дегенмен, жұмыс GREP ДК үшін сызықты дәлме-дәл салыстыратындықтан және егер сызық ДК болмаса, онда жолда сәйкестік бар. Егер сызық 444 GH, Фокс, PC, Linux болса, онда бұл сәйкес келетін болады, себебі жолда ДК болып табылмайтын, жолды іске қосатын таңбалар бар, мысалы '44'. Біз шынымен де мынаны білдіретін тұрақты өрнекке ие болғымыз келеді: "кейбір заттар Артында C жоқ P-ға ермейді, сосын кейбір басқа заттар". Осы тұрақты өрнекті құру күрделі міндет болып табылады. Оның орнына, біз ДК тұрақты өрнегін қолданып, -v параметрін gper пәрменіне қосар едік. Яғни, egrep -v 'PC' offices.txt

Бұл пәрмен 'PC'-ға сәйкес келетін әрбір жол үшін көрінеді, сосын файлдағы қалған жолдарын қайтарады. Өкінішке орай, бізде PCA немесе APC кеңсесімен жазба бар, бір компьютері бар немесе жоқтығына қарамастан бұл жол қайтарылмайды. Осылайша, біз ғимарат атында ДК келісуін болдырмау мақсатында "ДК" болу үшін тұрақты өрнекті реттейміз, сондықтан кеңейту пәрмені берілді:

egrep -v ' , PC' offices.txt

Тұрақты өрнектерді қолданудың басқа әдістері

gper / кеңейту беруден бізге метатаңбалардың тұрақты өрнектерінің толық спектірін қолдануға рұқсат етілген. Бірақ біз тек алмастыру таңбаларымен ғана шектелгенбіз. Егер топтық таңбалардың емес, тұрақты өрнектердің көмегімен құрылуы мүмкін үлгіге сәйкес келетін белгілі бір файлдарға арналған тізімдеме (немесе тізімдеме жинағын) табуды қаласақ, не істеу қажет? Мысалы, егер сіз құқықтары тек оған ғана қолжетімді барлық файлдарды атап шыққыңыз келсе не істейсіз? 6 тараудан рұқсаттар -l Ls пәрменінің көмегімен көрсетілетіндігін еске түсірейік. Оқуға ғана арналған файл рұқсат етулер арасында g-- қамтитын қолжетімділік құқығына ие болады (шындығында, егер файл gwxgwxg-- немесе g ----- орнатылатын болса маңызды емес немесе g—рұқсаттардың бір жерінде болғанға дейін ғана басқа нұсқа).

Ls үшін * "ештеңе" мәнге ие. Сондықтан, артында g-- "ештеңе" келетін "бірдеңе" дәлме-дәл мәнін қалаймыз. Сол жақ шкала -l * g - * осы тапсырманы біз үшін орындайды ма? Жоқ. Осы пәрменнің қалай жұмыс істейтіндігіне қарайық. Біріншіден, Bash интерпретаторы * g-- * белгілеуін өрістетеді. Бұл Bash интерпретаторының ілесімі екі сызықша, сосын бірдеңе бар g-ге ие ағымдағы директориядағы барлық аттарды алады. Мы-

салы, foxg – мәтін атауындағы g-- байланысты осы үлгіге сәйкес келетін болады. Сосын, олар LS пәрменіне берілетіндігі туралы файлдар тізімі алынғаннан кейін, ұзын тізім көрінеді. Өкінішке орай, біз мұны дұрыс тәртіпте жасамадық, біз ұзын тізімді қалаймыз, сосын * g - * қолданғымыз келеді.

Біздің шешіміміз оңай. Ұзын тізімді алу үшін, біріншіден, біз Ls -l жасаймыз. Сосын кеңейтілген нәтиже береміз. Біздің нұсқаулығымыз Ls -l * | ағымдағы тізімдемеде барлық элементтердің ұзын тізімін алатындай 'r--' кеңейтуін беру болады, сосын біз g-- бар кез келген жолды іздейтін және соларды қайтаратын кеңейтуді беру үшін тізімге (жолдар сериясына) өтеміз. Бұл пәрмен ұзын тізімдегі g-- қамтитын кез келген тізімді қайтаратын болады. Егер foxg - мәтін аты бар файл болса, ол тіпті оның құқықтары g--сәйкес келмеген жағдайда да қайтады. Осыны қалай болдырмауға болады? Кеңейтулердің ұзын тізімдегі бірінші іс және файлдағы аты соңғысы болып табылатындығына назар аударыңыз. Біз неғұрлым дәл тұрақты өрнек жазып, оны жолдың басына сәйкес келуін мәжбүрлеу үшін ^ қоса аламыз.

Қолжетімділік құқықтары файл түрінен басталады. Ол үшін бізге қандай таңба алатынымыз маңызды емес, бірақ бұл тек бір таңба ғана болу керек. Біз кезеңді қолдана отырып, кез келген жалғыз таңбаны ала аламыз. Сосын біз be g, w, x тоғыз таңбадан немесе осы тоғыз шалакокс пен сандардан тұратынын күтеміз, біз кез келген g-- сәйкес келеміз. Осылайша, біз (g-- | [RWX -] {3}) тұрақты өрнегін қолданар едік. Бұл g-- немесе кез келген R, W, X комбинациясына және - үштен көп таңбаларға сәйкес келеді. Өкінішке орай, бұл біз үшін жұмыс істемейтін болады, себебі бұл RWX немесе gw-, тіпті --- сәйкес келуі мүмкін. Біз оның орнына (g - [RWX -] {6} | [RWX -] {3} g - [RWX -] {3} | [RWX -] {6} g--) ретінде өрнекті жаза аламыз. Мұнда біз g-- өрнектің ішінде көруді талап етеміз. Енді біздің пәрмен неғұрлым күрделі, бірақ ол g-- файл атында (немесе пайдаланушы атында немесе топтың атында) болатын сәйкестіктің алдын алады. 10.5 сурет шешімді көрнекілейді; дискреттілікке көмектесу үшін бос орындар "|" айналасына қойылады.

Сіз Ls -l комбинациялап, көлемі 0-ден көп файлдар, нақты пайдаланушыға тиесілі файлдар немесе осы жылы құрылған файлдарға тиесілі әр түрлі заттарға арналған кеңейтуді іздейді. Ls -l қолдана отырып, көлемі 0-ден көп файлдардың ұзын тізімін алу және кеңейтуді беру үшін сіз қандай тәсіл ойлап табар едіңіз? Бұл сұрақ осы тараудың есептерінде беріледі (22 және 23 сұрақтарды қараңыз).

Grep-тен тыс тұрақты өрнектер қолданатын бірқата басқа бағдарламалар

бар. sed бағдарламасы ағындар редакторы болып табылады. Бұл бағдарлама редактор арқылы файлмен тікелей айла-шарғы жасау қажеттілігінсіз файл құрамын редакциялау үшін қолданылуы мүмкін. Мысалы, сіз мәтіндік файлдың бірінші жолындағы бірінші сөзді алып шығарғыңыз келеді деп санайық. Сіз vi немесе Emacs-дегі файлды ашып, қолмен редакциялай аласыз. Немесе сіз SED бағдарламасын қолдана аласыз. SED-те тұрақты өрнек пен ауыстыру жолын көрсету қажет. Тұрақты өрнек сіз іздеген жолды іздейді. Ауыстыру жолы табылған жолдың орнына қолданылады. Сіз ауыстыруды дәлме-дәл көрсете аласыз немесе «жоғарғы регистр» немесе «төменгі регистр» секілді пәрмендерді қолдануға болады. Сіз элементті жоя аласыз.

SED құрал файлда үлкен әрі жылдам алмасуды жасау үшін өте пайдалы. Дегенмен, ол тұрақты өрнектердің берік түсінігін талап етеді. HTML-файлдан барлық HTML-тегтерін жою үшін бір қарапайым мысал. Тұрақты өрнекті "<. *> ' секілді анықтауға және оны 0 (немесе бос орынмен) ауыстыруға болар еді. Сондықтан <> белгісіндегі ештеңе HTML-тег болып табылады, бір пәрменді табуға және одан бәрін жоюға болады. SED үшін басқа қолдану – файлды қайта форматтау. Ақпарат жолдан кейін жолда сақталмайтын, әрбір элементті бөлу үшін үтір арқылы сақталатын файлды қарастырайық. Сіз үтірлерді (\t) табуляция таңбаларымен және/немесе (\n) жаңа жол таңбаларымен ауыстыра аласыз.

Белгілеуге арналған тағы бір бағдарлама – AWK. Бағдарламаның аты AWK бағдарламасын жазған үш бағдарламашының инициалдарын қамтиды. SED ауыстыру үшін жолдарды іздеп жатқанда, AWK өңдеу үшін жолдарды іздеуді орындайды. AWK-мен тұрақты өрнектер жұптары мен әрекеттерді көрсету қажет. Егер жол тұрақты өрнекке сәйкес келсе, онда жол көрсетілген әрекеттердің көмегімен өңделеді. AWK-ны қолданудың бір қарапайым мысалы – тұрақты өрнекке сәйкес келетін жолдың жекелеген элементтерін шығару. Егер сәйкестік табылған кезде кеңейтуді беретін болсақ, онда барлық жол шығарылады, бірақ AWK нені шығарғыңыз келетінін көрсетуге мүмкіндік береді. Осылайша, AWK бағдарламаның деректер қорына біршама ұқсайды, бірақ ол деректер қорына қарағанда анағұрлым мықтырақ болып табылады, себебі, келістіру шарты тұрақты өрнекке негізделген.

AWK-ны қолданудың басқа мысалы – сәйкес келген элементтердің математикалық операциялары. Мысалы, мәтіндік файл қызметкерлер үшін еңбекақы туралы ақпаратты қамтиды деп қарастырыңыз. Ақпарат ішінде жұмыскерлердің аттары, осы аптада жұмыс істелген сағат саны,

еңбекақы және салықтық ақпарат болады. AWK көмегімен біз төлеуге қажетті мерзімінен артық төлем сомасын есептей аламыз. Немесе әрбір жазбаны тексеріп, жұмыс істелген сағаттардың орташа санын есептеп шығаруға болады.

Тұрақты өрнектер VI және Emacs екеуіне де қамтылған. Жолды іздеу кезінде сіз таңбалы жолды көрсете аласыз, сондай-ақ тұрақты өрнекті де көрсетуге болады. SED секілді бұл қызығушылық тудыратын нақты жолдарды анықтауға мүмкіндік береді, сондықтан сіз оларды редакциялап, форматтауыңыз қажет.

Ақырында, тұрақты өрнектер көптеген бағдарламалау тілдеріне қамтылған. Тұрақты өрнек формасы 1960 жылдардың басында SNOBOL тілінде бірінші рет енгізілді. Дегенмен, 1987 жылға дейін болды, сосын тұрақты өрнектер Perl бағдарламалау тіліне енгізілді. Perl қуаттылығы тұрақты өрнектерді анықтауға және оларды айнымалы файлдарда сақтауға бағытталған. Perl пайдалы болғаны сонша, ол сценарийлердің веб-серверін қоса алғанда, көптеген интернет-қосымшаларды қолдау үшін қолданылатын тілге айналды. Содан бері тұрақты өрнектер PHP, Java, JavaScript, .Net, Python, Ruby платформалары мен Visual Basic қоса алғанда, жаңа тілдерге қосылды.

Біз осы тарауды тұрақты өрнектерге қатысты екі қалжынмен аяқтаймыз. "Егер сізде мәселелер болса және сіз оларды тұрақты өрнектерді қолдана отырып шешуді ойласаңыз, онда сізде екі мәселе бар".

Сұрақ: Бір тұрақты өрнекті басқасына жасауға не айтады?

А:.. *

11. Үдерістер мен қызметтер

Үдерісті басқару және үдеріс 4-тарауға қамтылды. Бұл тарауда пайдаланушының үдерісті бақылауы Windows 7 және Red Hat Linux пайдаланушы тұрғысынан жабылады. Үдерістерді іске қосу, мониторинг үдерістері, жоспарлау үдерістері және аяқталатын үдерістер. Әрі қарай тарауда операциялық жүйе ретінде қызметтің рөлі қарастырылады. Жалпы қызметтер Linux-те қызметтерді баптаула қарастырылады. Ақырында, жүйені бастапқы жүктеме (инициализация) кезінде қызметті іске қосуды автоматтандыру Windows 7 де, Red Hat Linux үшін де қарастырылады.

Осы тарауды оқуға арналған мақсаттар:

- Windows 7 мен Red Hat Linux-де үдерістерді іске қосу қадамын, бақылау/басқару және аяқтауды сипаттау.
- Windows и Linux тапсырмаларын, at, batch және crontab пәрмендерін жоспарлаушының көмегімен жоспарлау үдерісін енгізу.
- Операциялық жүйелерді қолдау қызметінің рөлін талқылау.
- Linux қызметтерін баптау үдерістерін сипаттау.
- Windows 7 мен Linux-те сервисті іске қосуды басқаруды талқылау.

Үдеріс туралы тарау 4-тарауға енгізілген. Бұл тарауда біз пайдаланушылар мен жүйе әкімшілерінің өздерінің үдерістерін қалай бақылайтындығын қарастырамыз. Сондай-ақ, біз жүйе әкімшісінің көмегімен басқаруға болатын, фондық үдерістер болып табылатын қызметтердің операциялық жүйесін (ОЖ) қарастырамыз.

Үдерісті іске қосу

Бағдарламаны іске қосу үшін бағдарлама орындалатын болуы тиіс, яғни, бағдарлама компьютерлік тілде болуы тиіс. Көптеген бағдарламалар жоғары деңгейлі тілде (мысалы, Java, C++) жазылғандықтан, бағдарламашы бағдарламаны жоғары деңгейлі тіл нұсқасынан компью-

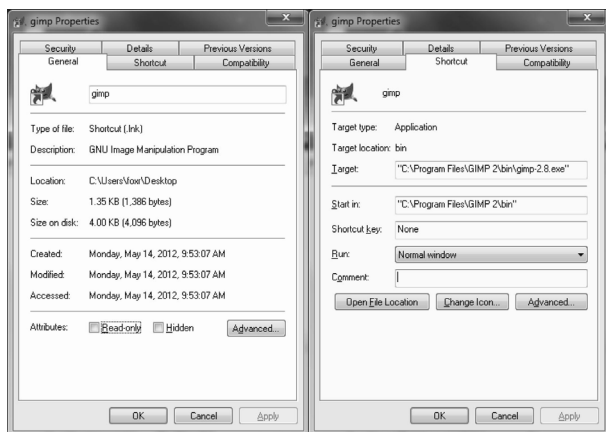
терлік тілге аудару қажет. Бұл әдеттегідей, бағдарламаларды – компиляторларды қолдану есебінен қол жеткізіледі. Компилятор бағдарламаның орындалатын нұсқасын білдіреді.

Сізге бағдарлама жұмыс үшін арналуы үшін әрбір бағдарламаға бағдарлама құру қажет болады (мысалы, Macintosh, Windows, Linux үшін жекелеген жинақ қажет).*

Кез келген үдерісті іске қосудағы бірінші қадам қолжетімді орындалатын бағдарлама нұсқасының болуы тиіс. Орындалатын бағдарлама файлдық жүйеде сақталады (әдеттегідей, қатты дискте, бірақ сондай-ақ оптикалық дискте, флэш-дискте және т.б. сақталған бағдарламаларды іске қосуға болады. Пайдаланушы бағдарламаны іске қосқысы келгенде, операциялық жүйе орындалатын бағдарламаны табуы тиіс. Көптеген операциялық жүйелерде пайдаланушы бағдарламаның бір немесе бірнеше мүмкін тәсілдемелер арқылы өтуі тиістігін көрсетеді.

Біріншіден, бұл таңбашалар болуы мүмкін. Таңбаша бағдарламаларға сілтемелердің графикалық көрінісін білдіреді (5-тараудағы сілтемелер). Таңбашаға екі рет басу ОЖ-ға бағдарламаны іске қосуға мәжбүрлейді. Қасиеттер сілтемесі файлдық жүйедегі файлдың орналасуын қамтиды. Windows операциялық жүйесінде сіз тышканның оң жақ батырмасын таңбашаға шертіп, Қасиеттерді таңдай отырып, таңбашаның қасиеттерін қарай аласыз.

Gimp бағдарламасына жылдам қолжетімділіктің мысалы ретінде 11.1 суретті қараңыз (графикалық редактордың ашық бастапқы коды бар): Таңбаша қасиеттерінің терезесінде бірнеше қосымша беттер бар.



11.1 сурет. Екі таңбашалар қасиеттерінің екі қосымша беті.

Қосымша бет таңбашаның орналасқан жерін қамтиды, ал Таңбашаның қосымша беті Target атты жолдағы орындалатын файлдың орналасқан жерін қамтиды. Таңбашаның өзі ағымдағы пайдаланушының жұмыс үстелінде сақталады, ал орындалатын файл Gimp \ Bin ішкі каталогындағы Program Files папкасында болады. Таңбашаға екі рет шерту операциялық жүйені мақсатты орналасқан жердегі сақталынған бағдарламаларды іске қосуға мәжбүрлейді. Редакциялаудың мақсатты терезесі – жағдай екендігіне назар аударыңыз, сондықтан сіз қажет болғанда орындалатын файлдың орналасқан жерін өзгерте аласыз. Қауіпсіздік қосымша беті де пайдалы ақпаратты қамтиды; ол таңбаша үшін файлдарға қолжетімділік құқығына ие (бұл рұқсаттар орындалатын файлдың рұқсаттарымен сәйкес келуі тиіс).

Бағдарламаны іске қосу үшін осы бағдарлама құрған деректер файлына тышқанмен екі рет шертуге болады. Әдеттегідей, аттардың кеңейтілуі (мысалы, docx, .pdf, .mp3) қосымшаларда көрсетіледі. Деректер файлын екі рет шерту ОЖ-ның тиісті бағдарламаларда көрінуіне әкеледі. Бұл бағдарламалық қосымша іске қосылады және файл қолданбалы бағдарламалық қамтамасыз етуде ашық.

Көптеген операциялық жүйелер бағдарламаның мәзірін береді. Көбісі таңбашалар секілді орындалатын файлдың мақсатты орналасқан жерін сақтау бағдарламасы мәзірінің тармақтары. Windows операциялық жүйесінде тышқанның оң жақ батырмасымен бағдарламаның атын, бағдарлама мәзірін және қасиеттерді таңдауды шерте отырып, 11.1 суретте көрсетілгендей терезені шақыртады.

Linux Windows секілді бағдарламалар мен таңбашалар мәзіріне ие. Дегенмен, Linux-тен (сондай-ақ, DOS) сіз командалық жолдан үдерісті іске қоса аласыз. Шындығында, жұмыс үстеліндегі, бағдарлама мәзіріндегі немесе командалық жолдағы белгіден бағдарламаны іске қосқанда олардың барлығы осылайша сол тапсырманы орындайды. Алайда, командалық жолдан сіз қосымша параметрлерді бере аласыз. Көптеген жағдайларда, таңбашаға арналған мақсатты орналасқан жері параметрлерді қамтымайды. Мысалы, Linux-те сіз VI -l someprogram.lsp көрсете аласыз. Опция -l Лисптегі бағдарламаларды редакциялауды қолдау үшін "lsp режиме"-дегі VI басталады. Сонымен қатар, пәрмендегі файл атын бере отырып, VI жүктелген файлды ашады.

Сіз бағдарламаны іске қосу үшін өзіңіздің пәрменіңізді жазып болғаннан кейін, операциялық жүйе барлығын өзіне алады. Біріншіден, ол файл жүйесінде сақталған орындалатын кодтың орналасқан жерін көрсетуі тиіс. Каталог тізімі бағдарламаның бастапқы орналасқан жерін (дисктік блогын) көрсеткішті қамтитын болады. Әрі қарай операциялық

жүйе тарту файлындағы орындалатын кодты жүктеуі және орындалатын кодтың бөлігін жадыға көшіруі тиіс (әдетте бірнеше бет). Егер жадыда бос калрлар болмаса, операциялық жүйе жою үшін бірқатар кадрларды таңдайды. Операциялық жүйе үдеріс үшін кестені құрады және үдерістің жағдайын сипаттайтын және оны кезекке қоятын (процессор жүктеуі мен үдеріс басымдылығына байланысты дайын немесе күтудегі) деректер құрылымы мен үдеріс сәйкестендіргішін (PID) құрады. Соңында процессор осы үдерісті алады және оны орындауды бастайды. Бұл басқа үдерістердің сол уақытта не орындап жатқанына байланысты баяу орындалуы немесе бірнеше миллисекундты қамтуы мүмкін.

Linux-те (және Unix) үдерістер – ата-аналық немесе еншілес (немесе екеуі). Бақса үдерісті бастайтын үдеріс ата-аналық үдеріс ретінде белгілі. Басқа үдеріс іске қосқан үдеріс еншілес үдеріс деп аталады. Мысал ретінде, егер сіз Windows терминалында болсаңыз және үдерісті бастау үшін пәрменді енгізсеңіз, мысалы, VI, онда үдеріс ата-ана болып табылады және бұл жағдайда VI үдеріс еншілес үдеріс болып табылады. Linux-те тек 1 үдеріс жүйені жүктеу кезінде құрылады, еншілес үдеріс емес. Барлық басқа үдерістер еншілес үдерістер болып табылады, бірақ ата-аналық үдерістер де болуы мүмкін. Келесі беттегі бүйірлік панель мен Process Execution-дегі бүйірлік панель үдерістер туралы толығырақ ақпарат береді және бір-бірімен ара қатынасын белгілейді.

Үдерісті жүзеге асыру

Үдерісті жүзеге асыру барысында пайдаланушы «орындау» үдерісін басқаруға айтарлықтай дәрежеде қатыспайды. Пайдаланушы үдерісті уақытша тоқтатуы, үдерісті жоюы, үдерісті алдыңғы орыннан соңғысына ауыстыруы мүмкін (немесе алдыңғы орындағы фонға) және үдеріс басымдылығын өзгерте алады. Керісінше жағдайда үдерісті орындау ОЖ қолында (немесе бақылауында) болады.

Сіз 4 тарауда айтылған бірінші орындағы үдеріс пайдаланушымен өзара әрекеттестікте болатынын есіңізге түсіре аласыз. Шын мәнінде бірінші орынға бір-біріне ұқсамайтын екі ұғым қойылады. Біріншіден, алғашқы орындағы үдеріс пайдаланушымен өзара әрекеттестікте болатын үдеріс болып табылады. Екі терезеде және графиктік пайдаланушы интерфейсі (ГПИ) Linux операциялық жүйесінде болады, ол барлық басқа үдерістердің «үстінен» жүзеге асырылады. Кейбір жағдайда тақырыпшаның басқа түспен белгіленгенін байқауға болады. Windows 7 сіз айырмашылығын байқай аласыз, себебі тапсырмалар тақтасындағы үдеріске арналған бетше біршама жарық, ал «үдеріс терезесінің өзі» оң жақ жоғарғы бұрышында қызыл түсті «X» белгіленеді. 11.2. суретте ал-

дыңғы орын фон үдерістерімен салыстырмалы түрде беріліп, тапсырмалар тақтасы мен терезе көрсетіледі (Word құжатының жоғары бұрышын қараңыз).

Үдерістердің алдыңғы орны ұғымы неғұрлым толық болады. Басым үдерістер дайындық калпында кезекте тұрғандар болып табылады. Яғни алдыңғы орында көп тапсырмалар процессорының үдерістері тұрады. Фондық үдерістер, яғни басымдылығы біршама төмен үдерістер процессор алдыңғы орындағы үдерістерден ГПИ босағанға дейін күтеді.

Жұмыс үстеліндегі үдерістің
алдыңғы орны

Group

тапсырмалар тақтасындағы үдерістің алдыңғы орны

11.2 сурет. Windows 7 фонына қарсы алдыңғы орын.

Бұл жерде біз бірінші кезекте алдыңғы орын ұғымының бұрынғы түсінігіне негізделеміз.

Фондық үдерісті алдыңғы орынға ауыстыру үшін графиктік интерфейсте оның бетшесін таңдаңыз немесе егер жұмыс үстелінде терезенің бір бөлігі көрінетін болса, терезені басу керек. Алдыңғы орындағы үдерісті артқы орынға ауыстыру үшін терезеде минималдау батырмасын басу немесе алғашқы орынға ауысу үшін басқа үдерісті таңдау керек.

Енді графиктік интерфейсте емес, Linux терминалының терезесінде бірнеше үдерістерді қосуды қарастырамыз. Сіз терминал терезесін пайдаланатын болғандықтан таңдауға қажетті бетше жоқ, үдерісті минимумге жеткізетін ешбір әдіс жоқ. Бұл терминал терезесінен бір уақытта тек бір ғана үдерісті қосуға болатынын білдіруі ықтимал. Дегенмен, олай емес. Терминал терезесінде алдыңғы орын және фон идеясы бірнеше қарапайым операциялардың бақылауында болады. Біріншіден, сіз үдерісті алғашқы орында іске қосу үшін жарлық атауын ғана енгізе аласыз, бұл жағдайда оқулықта көрген жарлықтар анықталады. Мысалы, VI <кіру> немесе Grep ... <кіру> немесе жоғарыдан <кіру> және аталған бағдар-

ламаны терминалдың осы терезесі үшін қосу керек. Үдерісті толықтай фондық режимде бастау үшін жарлықты орындап, одан кейін амперсанд (&) жүзеге асыру керек. Мысалы, егер, сіздің Grep тобыңыз көптеген жүйелік ресурстарды ала алатын болса, жүйе жұмысын бұзбау үшін сіз оны Grep ... &<іру> ретінде ресімдей аласыз. Бұл жағдайда ол фондық режимде жұмыс жасауы керек.

Сонымен қатар, сіз алдыңғы орыннан фонға немесе фоннан артқы орынға ауыстырып қосу мүмкін болады. Жарлықтар сәйкесінше, fg және BG болады. Алайда, егер үдеріс қазіргі кезде терминал терезесінде жұмыс жасайтын болса, сіздің fg немесе BG жарлығын беру үшін жарлықтар жолағына қол жетімділігіңіз болмайды. Жарлық жолағына қол жетімділік алу үшін осы терезеде Control + Z енгізу керек. Бұл үдерісті уақытша тоқтатады және басқару қайтадан сізге қайтады.

Мысалы, сіз жоғарыдағы жарлықты бересіз (жоғарыдағы келесі тарауда қарастырылады). Бұл интерактивті бағдарлама. Ол дисплейі бар терезені толтыралы және өзін жаңартады, сол себепті сіздің жарлық жолағына қол жетімділігі болмайды. Енді сіз Control + Z басқан сәтіңізде:

[1] + Stopped жоғары
көресіз.

Бұл жоғарыдағы 1 тапсырма тоқтап тұрған қалыпта қалатынын білдіреді. Жарлық жолағына тапсырманы енгізгенде сіз барлық іске қосылған және уақытша тоқтатылған тапсырмаларды көре аласыз. Егер тек бір жұмыс бар болса, FG теру арқылы осы жұмысты алдыңғы орында жаңартуға болады және бұл жұмыс фон режимінде жаңартылады. Жұмыс орнын жинақтағанда келесі түрдегі қалыпты көре аласыз:

> жұмыс орындары

[1] – Жоғарыда тоқтаған

[2] + Stopped VI

Мұнда екі аспалы жұмыс орындары бар. Егер сіз FG (немесе BG) теретін болсаңыз тапсырма таңдалады, бұл жағдайда + жақын уақытта орындалған жұмысты білдіреді. Дегенмен де екі санмен қандай тапсырманы алдыңғы орынға немесе фонға fg р немесе BG көмегімен ауыстыру керек екенін көрсетуге болады, мұндағы р тапсырма номері болып табылады. Бұл екі үдеріс интерактивті болып табылады, сондықтан оны артқы орынға ауыстырудың ешбір маңызы болмайды. Дегенмен де, бірін қайта жаңарту, уақытша тоқтату, екінші жағынан жаңарту, оны уақытша тоқтату және бірінші болып жаңарту оңай, мысалы, сіз құжат-

ты редакциялаған болсаңыз және біраз уақыттан кейін оны жүйенің өзі орындағанын көргіңіз келген жағдайда осы амал жасалады.

Басқа, мысал ретінде сізде екі қарқынды жұмыс уақыты мен редакциялауға арналған жұмыс бар делік. Сіз –іздеу жарлығын бересіз, ол барлық файлдық жүйе мен Гер үдерісін іздейтін болады, ол мыңдаған файлдарды іздеуі тиіс. Ол бірнеше секундтарға немесе минуттарға созылуы мүмкін. Осылайша, сіз VI құжатты да редакциялай аласыз. Сіз келесі үш жарлықты бере аласыз:

- > Табу ... басқару + g
- > BG 1
- > ... Кеңейтуді белгілеуді басқару + g
- > BG 2
- > VI

Енді сіз VI редакциялайсыз, бұл уақытта кеңейтілген фонда тауып алуға және қосуға болады. Мұнда қадамдар өз міндеттерін орындайды, алайда, келесі шамаларды оңай енгізуге болады

- > Табу ... &
- > ... & Кеңейтілген түрін белгілеу
- > VI

Бұдан әрі бұл тарауда оны PID қадағалау үдерісін бақылаудың тағы бір әдісін қарастырамыз. Тапсырма номері ретінде мұнда басқа мән қолданылады. Сіз тапсырмалар номерін арнайы тапсырмалар жарлығы көмегімен алуға болады. ИДП PS арқылы немесе топ-жарлықтар арқылы алуға болады (келесі тарауда қарастырылады).

Linux сіз «үдеріс басымдылығын» «жақсы» деп аталатын жарлық көмегімен басқара аласыз. Қабылданған жарлық «үдеріс сыпайылығына» жатқызылады. Linux сыпайылығы үдерістің басқа үдерістерге қатысты қаншалықты жақсы екенін көрсетеді. Қолайлы үдеріс CPU басқа үдерістерден артық бақылауда болады. Басқа сөздермен айтқанда озық үдерістің басымдылығы соғұрлым төмен болады. Қолайлы шама -20 (жақсы немесе ең жоғары басымдылық) және +19 (ең жақсы және ең төмен басымдылық) ауқымында болады.

Жақсы жарлық келесі форматты пайдаланып, жарлық жолынан үдерісті іске қосқанда беріледі:

Үдеріс-атауы жақсы -n мәні

Бұл үдеріс атауы – өзінің бастапқы мәнін белгілейтін шама түрінде орындалады. Егер сіз жақсыны пайдаланып жарлық беретін болсаңыз, бастапқы шама үнсіз келісім бойынша +10 анықталады. Үдеріс іске қосылғаннан кейін сіз оның сандық мәнін осы жарлықты пайдаланып өзгерте аласыз.

Мысалы, қандай да бір Foo үдерісін келесі жарлықтар арқылы іске қосамыз:

Жақсы Foo -n 0

Одан кейін келесі жарлықты орындап, Foo баптай аламыз:

жақсы Foo -n 19

Егер жақсы жарлығын үдеріссіз беретін болсаңыз.

FKO, г EXEC, обрабатываем

в Linux, мы используем такие термины, как вилка и Ехес. Что означают эти? это программы, используемые для запуска дочерних процессов, то есть, процесс будет вызывать на одну из этих программ, когда он должен генерировать дочерний процесс. разница в том, как команды реагируют.

вилка команда дублирует родительский и запускает новый процесс, как ребенок. ребенок имеет уникальный Pid, имеет родителя Pid своего родителя, и его собственные ресурсы, включая процессорное время. он получает свою собственную память, копирование памяти родителя для своих собственных изначально. ребенок также имеет доступ к файлам, открываемых родителем. как только вилка команда создала ребенка, родитель может либо продолжить выполнение, или войти в состояние ожидания. в последнем случае, родитель ждет, пока ребенка (или детей) прекратить перед возобновлением.

ехес команда похожа на развилке за исключением того, что порождает ребенок занимает место родителя. в этом случае, ребенок берет на себя Pid родителей и всех ресурсов. родитель прекращается, когда ребенок берет на себя, так как эти два процесса не могут совместно использовать один и тот же ФИД.

Үдерістің қалпы үдеріс туралы ақпарат болып табылады. Ол өзінің қалпындағы ресурстардан, сондай-ақ оның PID құралады. Windows және Linux сіздің үдерістеріңізді зерттеуге мүмкіндік беретін. Сіз, мысалы,

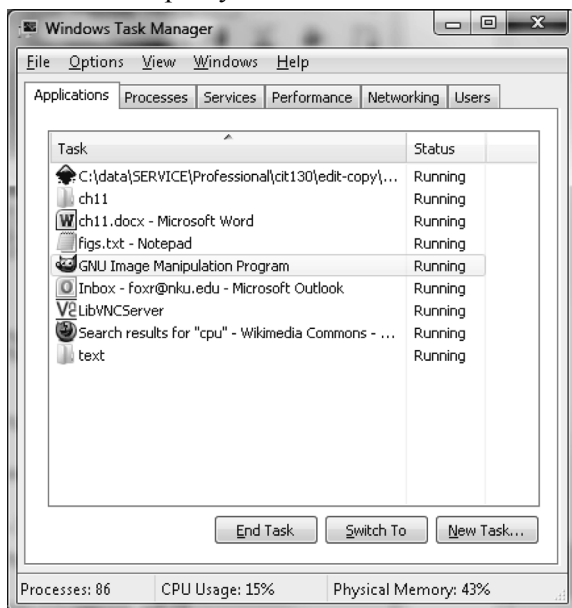
әрбір үдеріс қанша процессорлық уақыт алатыны, сіздің процессорыңыз қазіргі уақытта қаншалықты пайдаланылатынын және ағымдағы уақытта қанша виртуалды жады қолданылатынын біле аласыз. Сонымен қатар, өзіңіздің «басымдылық» үдерісіңізді ЦП уақытын қадағалай отырып бақылауға ала аласыз, мұнда негізінен көп тапсырма барысындағы үдерістерге баса назар аударылады. Мұнда біз ақпаратты өңдеу үшін процессорды зондтау үшін пайдаланылатын тәсілдерді қарастырамыз. Windows операциялық жүйесінде жүйенің өнімділігі және үдеріс қалпы туралы ақпарат осының барлығы Windows тапсырмалар диспетчері көмегімен қол жетімді болады. Іске қосылған қосымшаларға әдеттегі тұрғыдан қарауға болады. 11.3 суретте біз қазіргі кезде жұмыс жасайтын жеті пайдаланушылық қосымшалары бар екенін көреміз (үдерістердің өзінен басқа), барлық үдерістердің 74 (үдерістердің көбісі немесе жүйелік үдерістер пайдаланушылық қосымшаларын іске қосады). Процессорды пайдалануды қазіргі сәтте 0% құрайды және физикалық жады көлемі 28% (яғни жадының тек 28% ғана қолданылады).

Қосымшалар бетшесінде жана үдерісті бастау алдында үдерісті жою (End Task) немесе операциялық жүйені бір үдерістер екіншіге ауысу үшін қолдану керек. Шын мәнінде "Switch To"көмегімен сіз үдерісті фоннан алдыңғы орынға ауыстырасыз, ал ағымдағы басым үдеріс артқы орынға ауысады.

New Task ... таңдаған жағдайда жылжымалы терезе пайда болады. Осы терезеге орындалуы тиісті құрамдауыш сұранысының атауын енгізу керек. Жарлық орындалатын файлдың толық жолы түрінде болады, мысалы, "C:\Program Files\Internet Explorer\iexplore.exe". Үдерісті орындау барысында, ол администратор құқықтарымен жүзеге асырылады.

Қосымшалар бетшесінде тапсырмалар диспетчерінің ең маңызды қолданылатын орны сұранысқа жауап бермейтін тапсырмаларды жою болып табылады. Windows «жауап беруді тоқтатқан» тіркесін үдерістің тұйыққа тірелгенін, яғни жойылғанын білдірудің сыпайы әдісі болып табылады, алайда, операциялық жүйе оны өзі өлтіре алмайды. Үдерісті жою үшін оны үдерістер тізімінен ерекшелеп алып, Тапсырманы аяқтау – таңдау керек. Егер қосымшаның сақталмаған деректері бар болса, сізге қажет болған жағдайда бірінші кезекте ақпаратты сақтау ұсынылады. Үдерістер бетшесі Қосымшалар бетшесімен салыстырғанда неғұрлым көбірек ақпарат ұсынады. Біріншіден, барлық үдерістер тізімделген. Екіншіден, әр үдерісте «ағымдағы процессор мен жадыны пайдалануды» көре аласыз. Сонымен қатар, сіз үдерістің иесі кім екенін (оны кім бастағанын) біле аласыз. Windows операциялық жүйесінде сіз негізінен

төрт пайдаланушының бірін көретін боласыз: жүйе, жергілікті қызмет көрсету, желіге қызмет көрсету немесе меншікті пайдалану атауы.



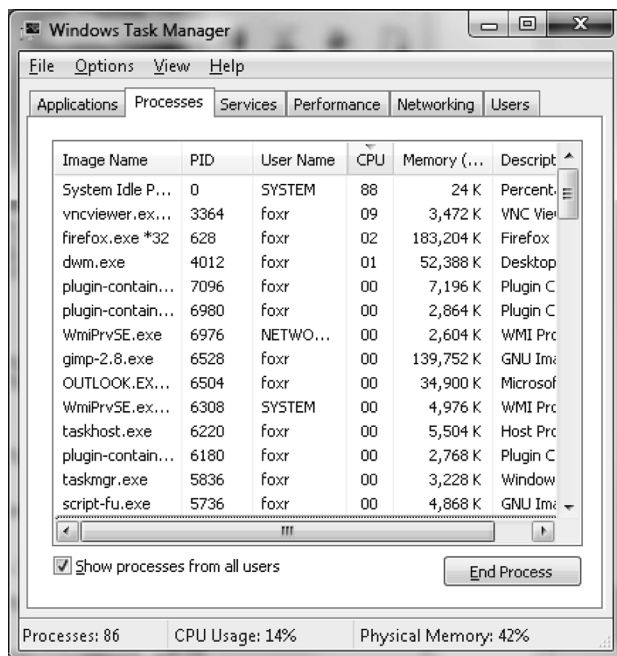
11.3 сурет.

Бастапқы үшеуі OS үдерісті бастағанын немесе үдерістің басқа бағдарламалық жабдықтаманы колдауды туындатқанын көрсетеді (11.4 суретті қар.).

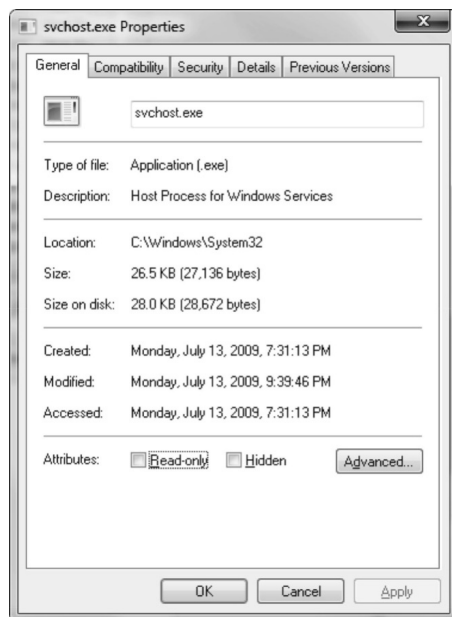
Үдерістер бетшесінде кез келген үдерістің бірін таңдау барысында, сіз оң жақ батырмамен оны басу және үдеріске «басымдылық» (оны көтеру немесе төмендету) және ұқсастық (қандай процессорлар немесе ядроларға үдерісті жүзеге асыруға рұқсат етілген) белгіленеді. Басымдылықтар алты шаманың бірімен шектеледі: төмен, қалыптыдан төмен, қалыпты, қалыптыдан жоғары және нақты уақытта. Үдерістің үнсіз келісім бойынша белгіленген мәні қалыпты болып табылады. Нақты уақыт ауқымында үдерістің маңыздылығы, оның өз міндеттерін орындауы және аш уақытта кейінге қалдырылмауы көрсетіледі.

Сіз, сонымен қатар, барлық үдерістерді аяқтай аласыз (ағаш үдерісінің соңы), үдерісті қадағалау немесе үдеріс экранына «қасиеттерін» шығару мүмкін болады. Үдеріс қасиеттерінің терезесі 11.5 суретте берілген. Мұнда сіз жалпы ақпаратты (файл типі, сипаттамасы, файл жүйесінде

орналасуы, өлшемі) көріп, қауіпсіздікті өзгерте аласыз, үдеріс туралы толық ақпарат алып және «үйлесімділік» үдерісін анықтай аласыз. Диспетчер қызметтері бетшесі Үдерістер бетшесімен бірдей болады, оның жалғыз ерекшелігі мұнда тек қызметтер ғана тізімделеді. Бұл қызметтер PID, сипаттама, қалып және топтар бойынша тізімделген. Кез келген қызметті басатын болсаңыз оны іске қосып немесе тоқтата аласыз. Сонымен қатар, барлық жүйелік қызметтер тізімін алу мүмкін болады. Біз қызметтерді толықтай қызметтер, Қызметтерді баптау және жүктеу кезінде қызметтерді құру барысында толықтай зерттей аламыз. Тапсырмалар диспетчерінің келесі екі бетшесі, желінің өнімділігі мен жұмысы жүйені ресурстарды пайдалану туралы нақты ақпаратпен қамтамасыз етеді. Өнімділік процессорды пайдалануды, жадыны негізгі қолдануды, виртуалды жадыны қолдануды және ағындардың, үдерістер мен басқарумен жұмыс жасайтын тұтқалардың ағымдағы санын қамтиды. Тұтқа ресурс типі түрінде болады және ашық файлдар мен жадыдағы айналымыларды қамтиды. Мәні жағынан тұтқа кестедегі нұсқаушы болып табылады.



11. 4 Windows тапсырмалар диспетчерінің үдерістері



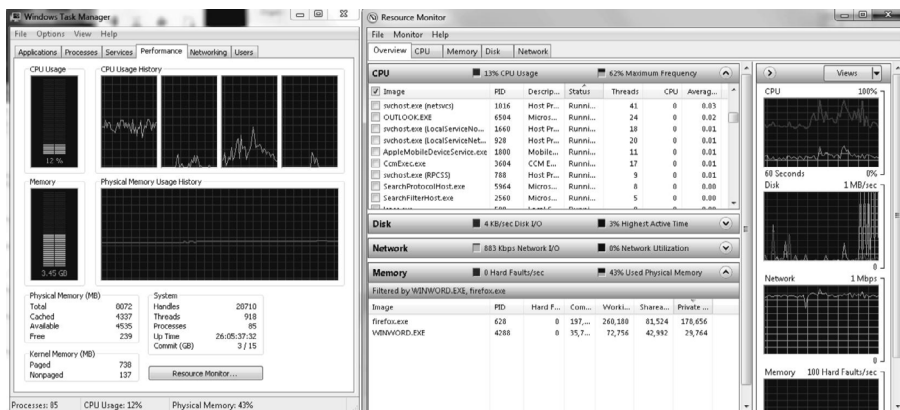
11.5 сурет. svchost.exe арналған үдеріс қасиеттері

Олардың нақты мәні дескриптор түрінде болады. Желі бетшесінде ағымдағы сымсыз жергілікті желі және жалғауларды пайдалану (бар болған жағдайда) көрінеді.

Өнімділік бетшесінде монитор ресурстары қызметін ашу мүмкін болатын батырма бар, ол процессор, жады, диск және желіні пайдалану туралы толық ақпаратты ашуға мүмкіндік береді. 11.6 суретте Өнімділік бетшесі мен «Ресурстар монитормы» көрсетілген, ол ресурстарды жалпы пайдалануды көрсетеді (ресурстар монитормының тек диск CPU немесе желіні пайдалануды оқшаулауға мүмкіндік беретін басқа да бетшелері бар екеніне назар аудару керек). «Users» бетшесінде барлық пайдаланушылар мен іске қосылған үдерістер көрінеді.

«Monitor» бетшесін пайдаланғанда сіз белгілі үдерістерді таңдай аласыз. 11.6 суретте firefox.exe (веб-браузер Mozilla Firefox) және WINWORD.EXE (Microsoft Word) қалай ерекшеленгені және оларды пайдалану жадыда көрсетілетінін байқауға болады. Сонымен қатар, бұл терезеде ақаулықтар санын көрсететін фильтр көрінеді (парақ ақаулықтары), олар ағымдағы секундта өтеді. Осылай сіз таңдалған үдерістерге арналған

«Disk» немесе «Network utilization» бетшесін таңдай аласыз.



11.6 сурет. «Өнімділік» бетшесі және монитор ресурстары терезесі

Linux өнімділігі туралы ақпарат алуға арналған бірқатар әр алуан түрлі құралдар бар. Үдеріс туралы ақпарат алудың неғұрлым дәстүрлі тәсілдері PS бағдарламасы арқылы (үдеріс қалпы) жарлықтар жолағында орындалады. PS жарлығының шатастыруы мүмкін бірқатар қызметтері бар, сол себепті ең жиі қолданылатын қызметтерді қарастырамыз.

PS жарлығын пайдалану өз алдында ағымдағы терезеде сіз пайдаланатын белсенді үдерістерді көрсетеді. Бұл ақпарат пайдаланушы үшін жеткіліксіз, себебі көрсетілмеуі мүмкін көптеген басқа да үдерістер бар, яғни олар басқа пайдаланушыларға тиісті болады (оның ішінде, негіз түбірі). 11.1 кестеде кең таралған қызметтер тізімі берілген. Көп жағдайда сіз PS жарлығында опцияларды ұсыну барысында пайдаланбайтын символдарға назар аудару керек. Бұл Unix қызметі, мұнда символдар талап етіледі.

11.1 кесте Linux PS. Жарлықтар мен қызметтер тізімі

a	Барлық үдерістерді көрсету
c	Жарлықтың нақты атауын көрсету
e	Нәтижесін көрсету
f	«parent»/ «child relationship» үдерістерін көрсету (ASCII-art

бағынышсаты түрінде)

l	Пайдаланушы анықтаған форматты көрсету
m	Барлық жалғауларды көрсету
o	Пайдаланушы анықтамаған форматты көрсету
p	ID үдерістерін таңдау
r	Қосылған үдерістерден шығу
S	«child process» ақпаратты қосу («parent» ақпаратымен бірге)
t	Терезе үдерістерін таңдау (tty)
T	Терминал үдерістерін таңдау
U	Белгілі пайдаланушы үдерістерін таңдау
u	Пайдаланушыға бағдарланған форматты көрсету
x	Терминал үдерістеріне жатпайтын үдерісті таңдау (tty)

Ең кең таралған PS пайдалану – бұл жарлықтардың өздері мен PS AUX жарлықтарын пайдалану болып табылады. PS AUX нұсқасы аса көп ақпаратты көрсете алады. Сол себепті нәтижені белгілі жолаққа сәйкес үдерістерді алу үшін GREP түрінде беру керек. Мысалы, PS AUX Grep foxg бұл жағдайда тек foxg тиісті үдерістерді ғана көрсетеді, алайда, барлық үдерістерді көрсетіп, терминалдың аталған терезесімен ғана шектелмейді. Мысалы, Bush көмегімен терминал терезесін ашу үшін Xterm енгізу керек, сонымен қатар, осы терминал терезесінен басқа жарлық беруге болады (ағымдағы файл компиляциясы үшін делік); осылайша, Bush компиляция жарлығының аналығы болатын XTerm аналығы түрінде болады.

```

PID TTY          TIME CMD
5031 pts/1      00:00:00 su
5034 pts/1      00:00:00 bash
5261 pts/1      00:00:00 ps

```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.1	2160	648	?	Ss	May29	0:01	init [5]
root	2	0.0	0.0	0	0	?	S<	May29	0:00	[migration/0]
root	3	0.0	0.0	0	0	?	SN	May29	0:00	[ksoftirqd/0]
root	4	0.0	0.0	0	0	?	S<	May29	0:00	[events/0]
root	5	0.0	0.0	0	0	?	S<	May29	0:00	[khelper]
root	6	0.0	0.0	0	0	?	S<	May29	0:00	[kthread]
root	9	0.0	0.0	0	0	?	S<	May29	0:00	[kblockd/0]
root	10	0.0	0.0	0	0	?	S<	May29	0:00	[kacpid]
root	168	0.0	0.0	0	0	?	S<	May29	0:00	[kqueue/0]
root	171	0.0	0.0	0	0	?	S<	May29	0:00	[khubd]
root	173	0.0	0.0	0	0	?	S<	May29	0:00	[kseriod]
root	237	0.0	0.0	0	0	?	S	May29	0:00	[khungtaskd]
root	238	0.0	0.0	0	0	?	S	May29	0:00	[pdflush]
root	239	0.0	0.0	0	0	?	S	May29	0:00	[pdflush]
root	240	0.0	0.0	0	0	?	S<	May29	0:00	[kswapd0]
root	241	0.0	0.0	0	0	?	S<	May29	0:00	[aio/0]
root	459	0.0	0.0	0	0	?	S<	May29	0:00	[kpsmouse]
root	488	0.0	0.0	0	0	?	S<	May29	0:00	[mpt_poll_0]
root	489	0.0	0.0	0	0	?	S<	May29	0:00	[mpt/0]
root	490	0.0	0.0	0	0	?	S<	May29	0:00	[scsi_eh_0]
root	493	0.0	0.0	0	0	?	S<	May29	0:00	[ata/0]
root	494	0.0	0.0	0	0	?	S<	May29	0:00	[ata_aux]
root	501	0.0	0.0	0	0	?	S<	May29	0:00	[kstripped]
root	510	0.0	0.0	0	0	?	S<	May29	0:13	[kjournald]
root	541	0.0	0.0	0	0	?	S<	May29	0:00	[kauditd]
root	574	0.0	0.3	3228	1664	?	S<s	May29	0:00	/sbin/udev -d

11.7 сурет PS жарлығы, PS AUX нұсқасы.

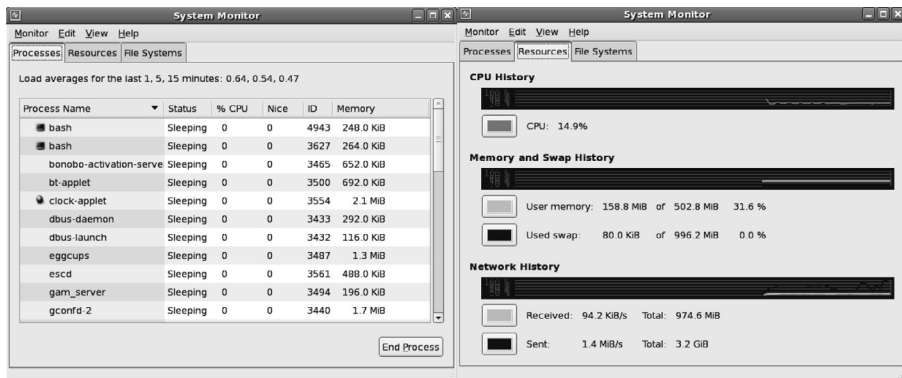
Қалып күй туралы ақпарат алуға арналған тағы бір жарлық жоғарыда орналасады. PS орын алған жағдайды суретке түсіргенде, сіз жарлықты интерактивті режимде жарлықты орындайсыз, ол өзін-өзі жаңартып әрбір 3 секунд сайын ең соңғы ақпаратты көрсетіп отырады. Бұл жарлық интерактивті болғандықтан ол терезеде қала береді және сіздің оны орындау барысында жарлық жолағына қол жетімділігіңіз болмайды. PS және тор арасындағы тағы бір айырмашылық, тор тек аталған сәтте белсенді болатын үдерістер ғана көрсетіледі. Сіз сізге тор жарлығының қызметтерін өзгертуге мүмкіндік беретін бірқатар параметрлер болып табылады. Негізінен өз жүйеңіздің тиімділігімен жұмыс жасайтын болсаңыз, сізге PS жұмыс жасау жеңіл болады және ол пайдалану барысында қолайлы болады. Тор жарлығынан шығу үшін Ctrl + C басу керек. 11.8 суретте осы жарлықты шығару көрсетілген.

Windows секілді Linux те жүйе жұмысына арналған графикаітік құралдарды ұсынады. Біріншіден, System Monitor құралы – бұл Windows тапсымалар диспетчері болып табылады. System Monitor үш бетшесі бар: үдерістерді көрсетуге, ресурсты пайдалануға және файлдық жүйеге арналған. Processes бетшесінде барлық үдерістер олардың дәрежесіне сәйкес тізімделген, оның ішінде, жүктеу, сыпайылық, UID және жадыны пайдалану қарастырылады. Осы жерден сіз үдерістерді тоқтатып немесе жалғастыра (ұйқы режимі) аласыз, үдерістерді жойып немесе олардың басымдылығын өзгертуге болады. Resources бетшесі процессорды, жадыны және желіні пайдалануды ағымдағы және соңғы уақытта көрсетеді. Файлдық жүйелер бетшесі ағымдағы белгіленген файлдық жүйені көрсетеді, ол df –k жарлығымен теңдес болады. 11.9 суретте үдерістер мен бетшелер ресурстары көрсетілген.

```
top - 05:08:54 up 6 days, 21:01, 3 users, load average: 0.52, 0.48, 0.45
Tasks: 130 total, 2 running, 128 sleeping, 0 stopped, 0 zombie
Cpu(s): 6.6%us, 3.6%sy, 0.0%ni, 83.4%id, 0.3%wa, 1.7%hi, 4.3%si, 0.0%st
Mem: 514908k total, 472176k used, 42732k free, 107128k buffers
Swap: 1020116k total, 80k used, 1020036k free, 205552k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3473	foxr	15	0	47856	9928	8208	S	10.0	1.9	14:33.72	vino-server
3086	root	15	0	34288	21m	10m	S	0.7	4.4	0:21.85	Xorg
3624	foxr	15	0	41252	12m	8560	R	0.3	2.5	0:01.84	gnome-terminal
1	root	15	0	2160	648	560	S	0.0	0.1	0:01.35	init
2	root	RT	-5	0	0	0	S	0.0	0.0	0:00.00	migration/0
3	root	34	19	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0
4	root	10	-5	0	0	0	S	0.0	0.0	0:00.44	events/0
5	root	10	-5	0	0	0	S	0.0	0.0	0:00.01	khelper
6	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	kthread
9	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	kblockd/0
10	root	20	-5	0	0	0	S	0.0	0.0	0:00.00	kacpid
168	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	cqueue/0
171	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	khubb
173	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	kseriod
237	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khungtaskd
238	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pdflush
239	root	15	0	0	0	0	S	0.0	0.0	0:00.17	pdflush
240	root	10	-5	0	0	0	S	0.0	0.0	0:00.14	kswapd0
241	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	aio/0
459	root	11	-5	0	0	0	S	0.0	0.0	0:00.00	kpsmoused
488	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	mpt_poll_0
489	root	20	-5	0	0	0	S	0.0	0.0	0:00.00	mpt/0

11.8 сурет. Файлдық жүйелер бетшесінің жарлықтары



11.9 сурет. Linux жүйелік монитормының екі түрі.

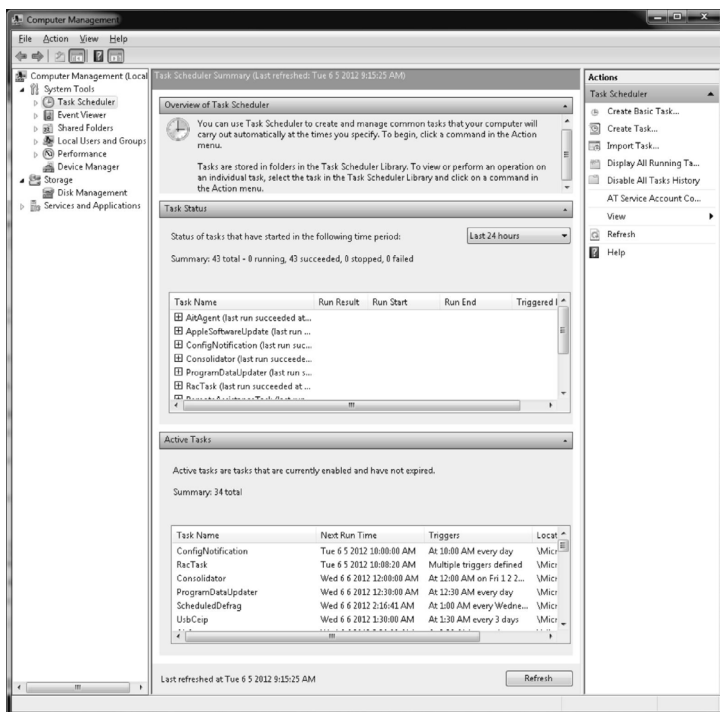
Сіз, Resource Monitor қосымша фондар мен қызметтерді баптау қызметінің графиттік интерфейсі арқылы ала аласыз. Сонымен қатар, қол жетімді болатын System Log Viewer графиттік интерфейсі де бар. Бұл іске қосылған үдерістер туралы тікелей ақпарат бермесе де, көптеген үдерістер Log files орындау уақыты туралы ақпаратпен қамтамасыз етеді. Журнал файлдары осы тарауда төменде қызметтермен қатар талқыланады. Configuration Manager Service секілді журналдарды қарау жүйесінің құралдары да сіз жүйеге түбегейлі кіргеніңізді талап етеді, бұл оларды пайдалану үшін қажет. Бұл жағдай System Monitor үшін талап етілмейді.

Операциялық жүйелер

Операциялық жүйе пайдаланушы үшін жоспарлауды жүзеге асырады. Жоспарлаудың әр алуан формалары бар. Мерзімді әсер ететін жүйелерде жоспарлау процессордың бағдарламаны орындау тәртібін анықтау үшін қажет болады. Пайдаланушылар негізінен Жинақты өңдеуді қолданбайтын болғандықтан, жоспарлаудың осы түрін біз біршама деңгейде ескермей қалдыра аламыз. Көп міндеттілікте, циклді жоспарлауда жалпы тәсіл осы үдерістерді жүзеге асыру болып табылады. Алайда, жоғарыда айтып кеткеніміздей, пайдаланушылар үдеріс басымдылығын айқын өзгерту және/немесе үдерістерді артқы орынға ауыстыру арқылы циклдік жоспарлауды өзгерте алады. ОЖ жүзеге асыратын жоспарлаудың тағы бір түрі үдерістердің күтіп тұрған кезегінен кесте бойынша орын ауыстыруы болып табылады. Дербес компьютерлерде бұл әдетте маңызды емес, себебі барлық үдерістер дайын үдерістер кезегіне орын

ауыстырады. Мұнда біз үдерістерді Windows және Linux жоспарлауды қарастырамыз, олар үдерістің дайындық қалпына ауысу уақытын көрсетеді.

Windows 7 тапсырманы жоспарлаушы бағдарламасын қамтамасыз етеді. 11.10 суретті қараңыз. Сіз жоспарланған әрекеттерді орындау үшін белгілей аласыз. Міндеттер күнделікті, апта сайын немесе ай сайынғы белсенділікпен немесе бір реттік құбылыс түрінде жоспарлануы мүмкін. Аталған жағдайлардың кез келген бірінде бастапқы уақытты көрсету керек. Мысал ретінде, сіз бағдарламада өзіңіздің вирусқа қарсы бағдарламалық жабдықтамаңызды өз компьютеріңіз әр жолы жүктелген сайын немесе әр күн сайын түнгі сағат 3 қосуды белгілей аласыз.



11.10 сурет. Windows операциялық жүйесі. Сіз сонымен қатар, резервтік көшіру және осы бағдарламаны жоспарлауды орындауға арналған бағдарлама жаза аласыз, ол аптасына бір рет орындалатын болады.

Linux жазба бағдарламаларын жоспарлаудың үш түрі бар. Осы барлық үш үдеріс фон режимінде іске қосылады; осылайша, үдерістер файлдар ақпаратын енгізуді талап етеді. Жарлық (s) тиісті қайта бағыттау жағдайында талап етілуі мүмкін. Үш үдеріс crontab болып табылады. Жиынтық файл жарлығы ЦП 80% төмен жүктегеннен кейін орындалатын үдерістерді жоспарлауға мүмкіндік береді. Жарлық іске қосу үдерістерін белгілі уақытқа жоспарлауға жағдай жасайды. Бұл жарлықтар жоспарланған үдерістерді бір уақытта іске қосады. Crontab жарлығы қайталанған жоспарлау үшін пайдаланылады. Осы жарлықтардың өзара қалай байланысқанын кейін талқылайтын боламыз, алайда, batch жарлығы үдерістерді іске қосуға мүмкіндік беріп, процессорды іске қосуға жағдай жасайтынын есте сақтау керек.

Екі файл да жарлықтарды қабылдай алады. batch (жинақ) жарлық форматы – бұл негізінен batch немесе batch –f өріс атауы. Әрбір үдеріс Linux жарлығы арқылы, оның ішінде, кез келген қажетті қайта бағыттау, мысалы, `cat foo1.txt foo2.txt foo3.txt | sort >> foo4.txt` арқылы орындалуы керек. Егер сіз –f қызметін пайдаланатын болсаңыз, жарлық сізді жолаққа қайта бағыттайды, ол `>` түрінде оқылады. Осы жолаққа жарлықты енгізіп, әр жарлықтан кейін `<enter>` басып отыру керек. Үдерісті аяқтау үшін `control+d` енгізеді. Осы амалдардың барлығы орындалғаннан кейін экранда келесі хабарлама пайда болады:

```
job 5 at 2012-03-01 12:38
```

ол жұмыс номері мен жарлық белгіленген уақытты анықтайтын болады (жарлық жұмыс жасауды бастайтын уақыт емес).

at сан жарлығына арналған уақыт индикаторының бірнеше форматтары болуы мүмкін. Бірінші тәсілде, сіз уақытты HH:MM форматында енгізе аласыз, мысалы, 12:35 t үдерісті 12:35 бастау. Күн мезгілін анықтамастан (танертең немесе кешке); осылайша, 12:35 – бұл танертең, 00:35 – бұл – кеш. Сізге минуттарын көрсетіп белгілеуге немесе көрсетпеуге де болады. Мұнда үш уақыт бағдарламаланған: түн, тал түс және бесін (16:00).

Жарлық үдерістері кестесінде келесі жолы іске қосу үшін белгіленген уақыт жазылады. Мысалы, егер сіз тал түсті көрсететін болсаңыз, ал қазір уақыт кешкі 1:35 болса, үдеріс келесі күні тал түсте орындалатын болады. Сонымен қатар, күнді 24 сағат бойы немесе одан кейін белгілеуге болады. Мерзімдері үш нотацияның бірімен қамтамасыз етіледі: MMDDYY, MM/DD/YY, немесе DD.MM.YY, мұнда MM – сан форматындағы ай (мысалы, 1 немесе 01 қаңтар және 12 желтоқсан), DD – ай саны, және YY- жылдың соңғы екі саны. Сонымен қатар, күннің орнына

күнді (ертең немесе арғы күні) және осы күндегі уақытты белгілеуге болады.

Сонымен қатар, күнін 24 сағат және одан кейін белгілеуге болады. Мерзімдер үш нотацияның бірінің көмегімен қамтамасыз етіледі.

Күн мен уақытты көрсетпеу үшін бүгінгі күнді пайдаланып, оған күндерді қосып отыруға болады. Сонымен қатар, сіз минуттар мен сағаттарды, күндер мен апталарды немесе айларды қосып отыра аласыз. Алайда, секундтарды қосып отыруға болмайды.

Сіз тапсырманы белгілегенде оны атқ көмегімен тексеру керек. Атқ нұсқаулығы сізге барлық күтіліп отырған тапсырмалар туралы қажетті ақпаратты көрсетеді. Atm жарлығы тапсырмаларды жоспарлау кезегінен өшіруге мүмкіндік береді. Тек бір ғана жұмыс бар болған жағдайда atm оны өшіреді.

Осы екім бағдарлама – жоспарлау бағдарламалары. atd қызметі жоспарлау кезегін қадағалайды, ол орындалған үдерістерді анықтау үшін қажет. Ол фон режимінде жұмыс жасайды және әр минут сайын ағымдағы уақыт мен тапсырма уақытын салыстырып отырады.

Басқа Linux жоспарлаушы crontab деп аталады. Бұл бағдарлама қайталанған тапсырмаларды жоспарлау үшін пайдаланылады, мысалы, әр дүйсенбі күні сағат түнгі бірде немесе әр айдың бірінші жұлдызында қайталанатын тапсырмалар. Ол қандай да бір дәрежеде бірінші жоспарлаушыға ұқсас, бірақ бұл күрделірек. Біріншіден, cron файлын құру керек. Cron файлы екі затты анықтайды: уақыт/күн және белгіленген үдеріс. Уақыт/күн бес саннан немесе * құралады. Бес мән келесі тәртіпте болады: минут, сағат, күн, ай және апта күні, ол тапсырманы кесте бойынша орындау үшін керек. Минут бұл 0-59 дейін аралықтағы сан, күн 0-23 аралығындағы әскери уақытты пайдалану, ай күні 1-31 дейін, ай 1-12 дейін және апта күні 0-6 дейін (мұндағы 0 – жексенбі, 6 сенбі) болады.

Уақытты көрсетудің кейбір мысалдары төменде берілген (crontab файлының бір бөлігі):

15	3	1	*	*	~~date~~ first month, every at 3:15 am
0	14	*	*	0	—every Sunday at 2 pm
30	*	*	*	*	~~a~~ every hour half past


```

0 0 12 31 * 12:30, 1:30, 2:30)
0 * * * * midnight December 31, at
0 * * * * every hour every day

```

-Әр айдың бірінші күні, таңертеңгі 3:15 сағат

-Әр жексенбі күндізгі сағат 2

-Әрбір бір жарым сағат сайын (мысалы, 12:30, 1:30, 2:30)

-Әр 31 желтоқсан сайын түнде

-Әр күні әр сағат сайын

Егер күн мен апта күнін (күн және ай) бір мезгілде белгілейтін болса, мысалы:

```
0 0 15 * 0
```

Әр 15 күн сайы және әр жексенбінің түнгі белгіленеді.

Егер бірнеше нақты күндер мен апта күндерін белгілеу қажет болса, бірнеше уақыт кезеңін белгілеу керек болса, сіз әр уақыт санаты үшін бірнеше жазбаны тізімдеп шыға аласыз, олар үтір арқылы бөлінеді. Мысалы:

```
0,30 0,6,12,18 * * *
```

Күн сайын орындалатын тапсырмаларды 12:00, 12:30, 6:00, 6:30, 12:00, 12:30, 18:00, 18:30 кешкі уақыттармен белгілей аласыз. Басқа нұсқа, егер сіз барлығы белгілі уақыт аралығында өткенін қалайтын болсаңыз, мысалы әр 15 минут сайын, ол /15 көмегімен белгіленеді, мысалы:

```
0/15 12 * * *
```

Әр күндегі 12:00 ол түнгі сағат 01:00 дейін әр 15 минут сайын қайталанбайтын уақыт.

Бұл аралықты келесі түрде көрсетуге болады:

```
0 0 1-5 * *
```

Әр айдың 1 және 5 күндері түнде белгіленетін болады.

Crontab файлының екінші жартысы – бұл орындалуы тиісті жарлықтар тізімі. Жарлықтардың бір жолақтағы файлдағы қалай орналасуы керек екеніне байланысты сіз бір жарлықты тізімдеп немесе скрипт жазасыз. Мысалы, жазба:

```
0 0 1 * * ./somescript -a < foo.txt
```

Орындауды `-a < foo.txt` әр айдың бірінші күніндегі түнге белгілеу керек. `script somescript` ағымдағы каталогта болады және опциялық және кіріс белгілерін `foo.txt` алады.

Сіздің әрекеттеріңіз шектеулі болатын себебі әрбір жоспарланған тапсырмада `crontab` жоспарланған жазбада бірнеше файлдармен жұмыс жасауға мүмкіндік беретіні, олар бір ғана жолақта жоспарланады. Осылайша, көптеген үдерістерді әр түрлі уақытта бір ғана `crontab` файлын пайдаланып жоспарлауға болады. `Crontab` файлындағы келесі жазбаларды қарастырамыз:

```
0    0    *    *    0    scripta c k u p _
0    0    *    *    2    scripte a n u p _
0/10 *    *    *    *    ./who_report
```

Бұл резервтік көшіру бағдарламасы әр жексенбі сайын түнде қосылғанын талап етеді, ал скрипт тазалау әр сейсенбі күндері түнде жұмыс жасайтын болады, жүйеге кімнің кіргені туралы хабарлама әр күн сайын әрбір 10 минутта жұмыс жасап тұратын болады.

`Crontab` файлының жазбасы тапсырмаларды `crontab` бірге жоспарлаудың алғашқы қадамы болып табылады. Екіншіден, `crontab` үдерісінің өзі берілуі керек. `Crontab` үдерісін `CRONTAB -f` файл атауы жарлығының көмегімен шақыруға болады. Қосымша `crontab` опциясы `-u` болып табылады және бұл пайдаланушы атауы. Бұл түбегейлі каталогқа ауысқанда және `crontab` берудің алдында әсіресе пайдалы болады. Егер `crontab` жұмыстары түбегейлі каталог ретінде жүргізілгенін қаламайтын болсаңыз түбегейлі қол жетімділік үшін қажетті үдерістерді білу керек болады.

`atq` және `atrm` мағдайындағыдай сіз кезекті `crontab` тапсырмалар кезегін тексеріп, оны өшіріп тастай аласыз. Оны `crontab -l` және `crontab -g` көмегімен жүзеге асыруға болады, тізімге кіріп, күтуде тұрған тапсырмаларды өшіру керек. Сіз ең соңғы `crontab` жұмысын редакциялау үшін `crontab -e` пайдалана аласыз. Іске қосу және жиынтық тапсырмалар үшін `atd` қалай пайдаланылатынына тәуелді `cron` мұнда `crontab` жұмыс орындарын іске қосу үшін қолданылады.

Соңғы ескерту, жүйелерді қорғау үшін сіз қандай пайдаланушылар `crontab` пайдалана алатынын және қайсылары пайдалана алмайтынын көрсете аласыз. Осындай қол жетімділікті қадағалайтын бірнеше файлдар бар: `/etc/at.allow`, `/etc/at.deny`, `/etc/cron.allow`, және `/etc/cron.deny`.

Токтату үдерісі

Үдерістерді тоқтатудың бірнеше тәсілдері бар. Біріншіден, ол орындауды аяқтауы мүмкін. Екіншіден, сіз оны тоқтатуды өз бетіңізше таңдай аласыз. Үшіншіден, бағдарлама «жойылуы», яғни оны орындауды тоқтату керек. Үдеріс өздігінен аяқталған жағдайда, ол орындауды аяқтағаннан кейін пайдаланушының шешімді өзі қабылдауының қажеттілігі болмайды. Үшінші жағдайда негізінен пайдаланушымен кері байланыс болады, ол тоқтату қателігі қалай туындағанын көрсету үшін керек. Мысалы, Windows операциялық жүйесінде сіз қателік туралы жылжымалы терезені байқай аласыз. Linux қате аяқталған бағдарламалар негізінен core dump файлын құрайды. core dump файлы жұмыс жадысынан үдерістерді алып тастайтын «қоқыс орны» түрінде болады. Бағдарламалаушы мәселенің шешімін табу үшін core dump файлын зерттеп қарастыруы керек.

Бұл шешімін табудың екінші мүмкіндігі. Үдеріс белсенді болады, алайда, енді аяқталуына қатысты ешбір қадам жасамайды. Бұл тұйыққа тірелген жағдай болуы мүмкін (4 тарауды қар.).

Windows операциялық жүйесінде үдерісті тоқтату үшін терезені жабу жеткілікті болады. Жауап қатпайтын үдерістер үшін бұл іске жарамауы мүмкін, сол себепті тапсырмалар диспетчерін шақырып, қосымшаны немесе үдерісті осы графтік пайдаланушы интерфейсі арқылы алып тастауға болады. Үдеріс терминал терезесінде (Linux) немесе жарлық жолағының терезесінде (Windows) қосылады және оны тоқтату үшін Ctrl + C басып көруге болады. Linux үдерісті таңдау үшін ресурстар мониторын да қолдануға болады. Сонымен қатар, сіз жарлық жолағынан жарлықты өшіріп тастау үшін оны пайдалана аласыз. Біріншіден, сіз ps жарлығы көмегімен PID үдерісін алуыңыз керек. Өшіру жарлығы – бұл PID деңгейі. Ең жоғарғы деңгейі -9 мүмкін болған барлық жағдайда қолданылуы керек. Killall жарлығы барлық белсенді үдерістерді өшіреді. Killall пайдалану үшін ол түбегейлі каталогта болуы керек.

Сіз ақпаратты іске қосылған үдерістер арқылы іздеуіңіз ықтимал, бұл қандай үдерістердің жұмыс жасап отырғанын көру үшін қажет болады. Бұл Windows және Linux үшін де дұрыс тактика. Бұл таңба тапсырмалар тақтасына тіркестірілмеген, бұл үдерістің осы уақытқа дейін іске қосылмағанын көрсетеді. Қызықты тарихи анықтама MIT 1973 жылы өздерінің IBM 7094 мейнфреймдерін жапқанда олар 1967 жылы іске қосылған басымдылығы төмен күту үстіндені үдерісті тапқан. Бұл оқиғаның қаншалықты шындық екені белгісіз.

Барлық жүйені сөндіру маңызды қадам болып табылады. Сізге компьютерді белгілі бағдарламалық жабдықтаманы орнатқаннан кейін (әсіресе операциялық жүйені түзеткеннен кейін) сөндіру керек болады. Бұдан

басқа, Windows 7 операциялық жүйені жаңарту барысында оны жауып, қайта жүктеуді талап етеді. Linux өзінің ОЖ жаңартуды қайта жүктеусіз орындай алады.

Windows операциялық жүйесінде кейде ОЖ қайта жүктеудің маңызы зор (мысалы, аптасына бір рет делік), ол жүйеде орын алуы мүмкін коррупцияға байланысты болады. Linux Windows салыстырғанда неғұрлым тиімді болғандықтан, оны қайта жүктеу қажет болмайды.

Сіз компьютерді қалай сөндіресіз? Кейде жай ғана сөндіру батырмасын басу арқылы сөндіргіңіз келмей ме. Windows және Linux сөндіру реттілігі болады. Оны барлық үдерістер мен қызметтер тиісті түрде жабылып, барлық файлдар сақталуы үшін дұрыс реттілігіне сәйкес жүзеге асыру керек. Егер сіз дұрыс жасамасаңыз деректер файлдары мен жүйелік файлдар істен шығады.

Windows операциялық жүйесінде жұмысты аяқтау үдерісі Іске қосу батырмасын басудан басталады. Жүйені сөндіру нұсқалары: жүйені қайта жүктеу (қайта жүктеу), жүйеден шығу, ағымдағы пайдаланушыны күту, күту режимі және ұйқы режимі. Ұйқы режимі энергия үнемдеу режимі болып табылады, бұл резервтік көшіруді бастауға мүмкіндік береді. Hibernate неғұрлым ұзақ режим болып табылады.

Linux сізге графиктік интерфейсті пайдалана аласыз немесе жүйені жарлық жолы арқылы сөндіруге болады. Бұл бізге «жеңілдікті кезеңді» береді, себебі барлық пайдаланушылар жүйеден уақыт бойынша шығуы мүмкін. Жарлық жұмысын аяқтау -h 10 жұмысты аяқтау және жұмысты тоқтату үшін 10 минут қажет екенін білдіреді. Сіз пайдаланушылар мен үдерістерге хабарлама жіберіп, оларға жұмысты аяқтау туралы ескерте аласыз. Хабарлама «жүйені 10 минут ішінде сөндіру туралы ескерту – Барлық үдерістерді жабу және шығу!» түрінде болуы мүмкін. Опция –г жүйені сөндіреді және оны қайта жүктейді.

Сервистер

Қызмет ОЖ бағдарламалық жабдықтамасының бір бөлігі түрінде болады, ол белгілі қызмет типтерін қамтамасыз етеді. Бұл дыбыс пен кез келген басқа құрамдауышты операциялық жүйеде жүзеге асырады. Бірнеше айырмашылықтары бар. Біріншіден, қызмет операциялық жүйе ядросын құрайтын бағдарламалар жұмыс жасап тұрғанда іске қосылуы немесе тоқтатылуы (немесе уақытша тоқтатылуы) мүмкін. Екіншіден, қызмет бағдарламалық жабдықтаманың агент-қосымшаларының кез келген санымен жұмыс жасауға бапталған. ОЖ ядросы пайдаланушы-

ның бағдарламалық жабдықтаманы басқару сұраныстарын өңдейді. Үшіншіден, қызметтер фон режимінде жұмыс жасайды, бұл жағдайда пайдаланушы араласпайды. Қызметтер фон режимінде жұмыс жасаған-дықтан қызметтер көп тапсырмалар жүйелері үшін қажетсіз болады.

Қызметтер жүктеу барысында іске қосылуы мүмкін, олар қандай да бір жоспарлау жүйесі негізінде іске қосылады немесе оларды жүйелік администраторлар (тоқтап қалған жағдайда) іске қосуы мүмкін. Linux қызметтерді көбінесе *daemons* (демонс деп айтылады) деп атайды, олардың атаулары *d* аяқталады, мысалы, *daemon* қадағалауымен жоспарлау жарлығы ATD. Windows операциялық жүйесінде олар Windows Services ретінде белгілі, алайда Linux айырмашылығы Windows кейбір қызметтері қалыпты үдеріс ретінде жұмыс жасап, артқы орын тапсырмаларын орындай алады.

ОЖ қызметтерінің артықшылықтарының бірі жүйелік администратордың ортаны сервистер жұмысы мен жұмыс жасамайтын қызметтерді ескере отырып, бе йімдей алатыны болады.

Бұдан басқа, жүйелік администратор қызметтің қалай жұмыс жасайтынын қадағалай алады. Жүйені іске қосқанда ол конфигурация файлдарын оқиды. Конфигурация файлдарын өзгерту арқылы осы қызметтің жұмысын бақылауға болады. Қызмет баптауларында Linux конфигурация файлдарын қарастыруға болады.

Қызметтердің бар болуының тағы бір артықшылығы қызметтер орындалған жағдайда олар фон режиміне орналастырылады. Мысалы, қызмет кезекті күтпейді және процессор уақытын пайдаланады.

Windows операциялық жүйесінде қызметтердің б операциялық жүйесінде қызметтердің бірнеше ерекшеліктері болады. Біріншіден, олар артықшылығы аз режимде жұмыс жасайды. Мысалы олар жергілікті қызмет немесе желілік қызмет шоттары бойынша жұмыс жасауы мүмкін. Бұл сервиске қосымша қол жетімділік құқықтарын береді. Бұдан басқа, Windows қызметтер жалпылама түрде және олар басқа қызметтерге ықпал ете алмайды.

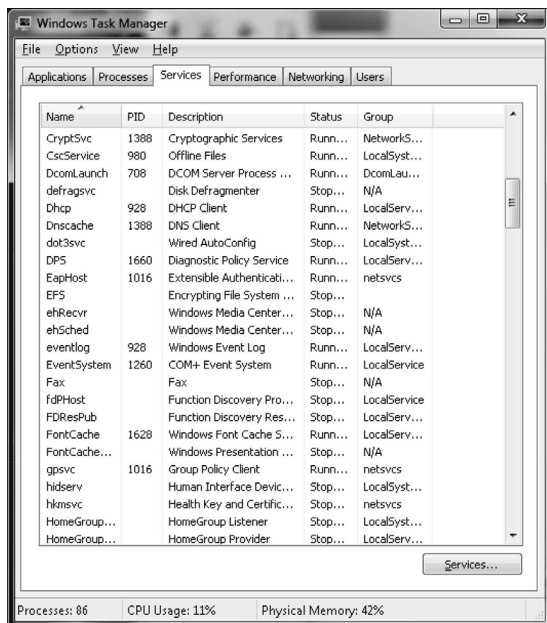
Linux көптеген қызметтер баптау үдерісінде пайда болады, ол Linux іске қосылғаннан кейінгі бірінші үдеріс болып табылады (Жүктеу барысында қызметтер құру тарауын қар.). Жүйе іске қосылғаннан кейін жүйелік администратор қызметті талап етуі бойынша кез келген уақытта қоса алады (немесе тоқтатады).

Қызметті сервермен шатастыруға болмайды, дегенмен, олар өзара байланысты. Біз барлық серверлер қызмет көрсете алады, ал барлық қызметтер сервер болып табылмайды деп айта аламыз. Сервер БЖ ірі

ауқымды бөлігі болып табылады (көптеген бағдарламалар ықтималдығы). Біз серверлерді толықтай 13 тарауда қарастыратын боламыз.

Көп жағдайда Windows және Linux қол жетімді қызметтер саны бар. Windows операциялық жүйесінде белсенді қызметтерді қарастыруға, оларды іске қосуға және тоқтатуға болады. 11.11. суретте Тапсырмалар диспетчеріндегі қызметтер бетшесі берілген. Сервисті таңдау оны іске қосуға және тоқтатуға мүмкіндік береді.

Мысалы, 11.11. суретте Netlogon, KeyIso және PlugPlay секілді қызметтер VaultSvc, EPS және bthserv қызметтер тоқтап тұрған уақытта қалай жұмыс жасайтыны көрсетілген. Сипаттама бағанында қызметке қысқаша сипаттама беріледі. 11.11. суретте бірнеше үдерістер 564 үдерістен бастау алған. Бұл негізінен lsass.exe бағдарламасы болып табылады. 708 үдеріс Power Services және PlugPlay бастаған және svchost.exe болып табылады. ИДП үдерісін алу үшін үдеріс бетшесін басып, Бағанды таңдау керек. Осы жерден қарау үшін қандай бағандар қол жетімді болатынын таңдауға болады. PID үнсіз келісім бойынша қол жетімсіз болады, алайда сіз оны көрінетіндей етіп жасай аласыз.

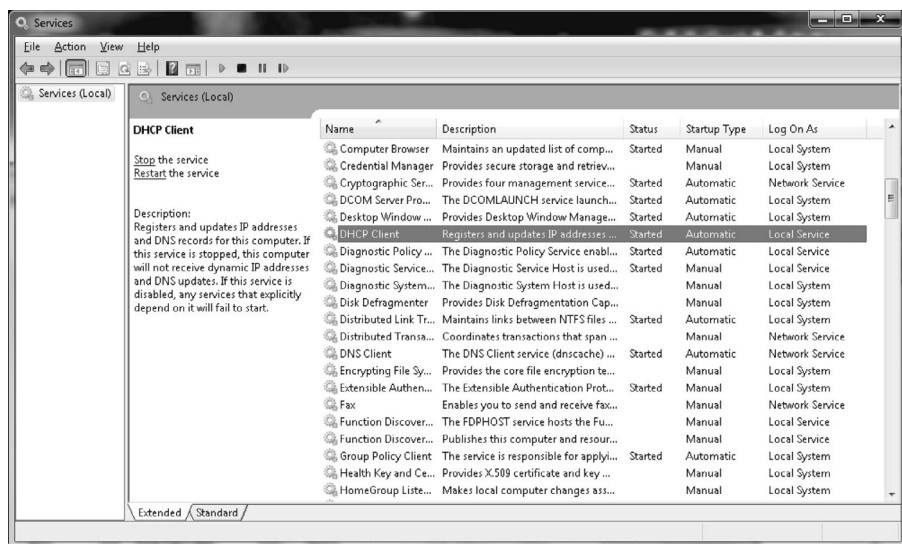


11.11 сурет. Тапсырмалар диспетчері қызметі бетшесі

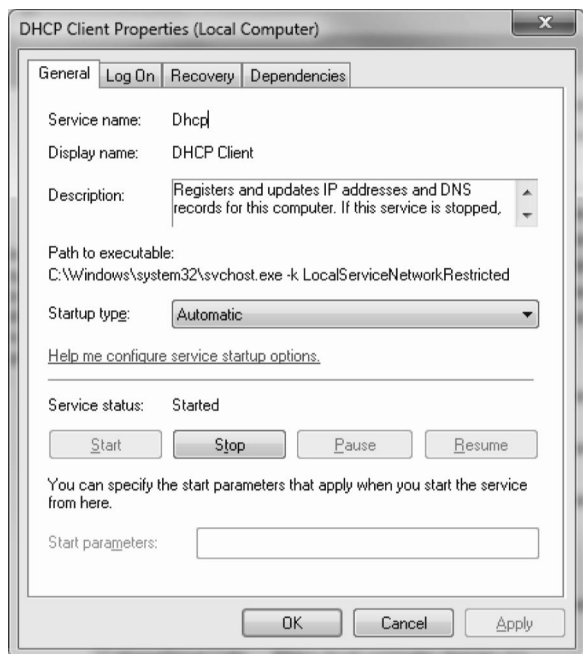
Қызметтер бетшесінің төменгі бөлігіндетерезе тапсырмалары менеждерінде Services батырмасы бар. Аталған батырманы басу қызметтер терезесін іске қосады (11.12 суретті қар.). Қызметтер терезесінен тышқанның оң жақ батырмасын басу арқылы Қасиеттер тармағын таңдауға болады (немесе ерекшеленген Қасиеттер), ол қызметтер қасиеттерін көрсетеді. 11.13 суретті қар. Бұл терезеде 4 бетше бар: Жалпы, Жүйегену, Қалпына келтіру және Тәуелділік.

Жалпы бетшесінде сервис атауы, оның файлдық жүйесіндегі орналасуы берілген (DHCP-клиент жағдайында ол автоматты түрде жүктеуден кейін іске қосылады). DHCP-клиент кез келген параметрлермен іске қосыла бермейді.

Іске қосу типінде төрт нұсқа бар. Автоматты (кейінге қалдырылған іске қосу), автоматты, қолмен, шектелген физикалық мүмкіндіктермен. Олар Windows Vista енгізілген жағдайда айқын болады. Барлық автоматты қызметтер жүйені жүктегеннен кейін қажет болады.



11.12 сурет Windows қызметі.



11.13 сурет. DHCP Терезе мүмкіндігі

Қызметтер маңыздылығы дәрежесі бойынша жіктеледі. Басымдылығы неғұрлым төмен қызметтер автоматты түрде жүктеу барысында іске қосылады. Ол « кейінге қалдырылған іске қосу» түрінде скрипттерді баптау басталғанға дейін белгіленеді. Мысал ретінде Windows іздеу және Windows жаңарту үшін қызметтер осы параметрді пайдаланады және оларға тікелей қызмет көрсету қажет болмайды.

Қызметтерге жүйелік және желілік ресурстарға қол жетімділік қажет болуы мүмкін. Егер бұл талап етілетін болса, онда логикалық енгізу керек болады (есепті жазба атауы және құпия сөз). Бұл жағдайда Жүйеге кіру түрінде белгіленеді (сіз оң жақ шеткі бағанда, Log On As қызметтерді көре аласыз). Мысалы, DHCP жүйеге жергілікті қызмет атауымен енгізіледі. Басқа нұсқалары Network Service және Local System болып табылады. Жергілікті қызмет пен Желілік қызмет есепті жазба журналы мен құпия сөзді талап етуі мүмкін.

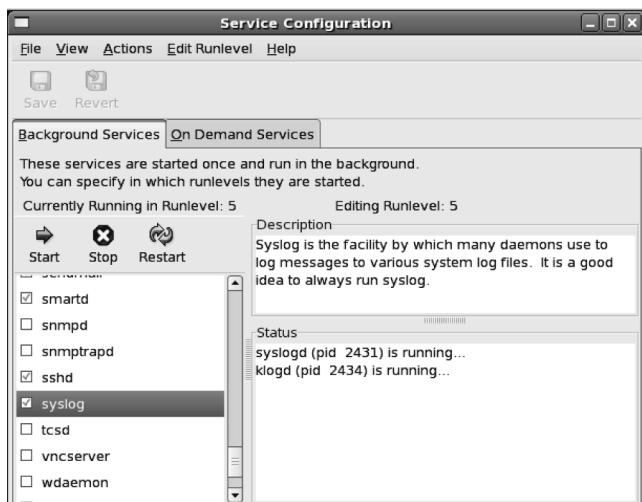
Қалпына келтіру бетшесінше ОЖ қызметті жүзеге асыра алмаған жағдайда қалай қалпына келуге тырысатыны көрсетіледі. Windows ОЖ

істен шығудың үш түрі болады: бірінші істен шығу, екінші істен шығу және одан кейінгі сәтсіздік. Бетшеде әр деңгейдің келесі элементтері болады: «ешбір амал қолданбау», «қызметті қайта жүктеу» (бұл жағдайда бағдарлама мен оның кез келген параметрлері көрсетіледі) және «компьютерді қайта жүктеу». Қайта жүктеу алдындағы кідіріс Lable басқару барысында да орын алуы мүмкін. Бұл шама минутпен белгіленеді (0 жоғары).

Тәуелділік бетшесінде қызметтер тізімделген, оларда бұл қызметтер аталған қызмет түріне тәуелді тәуелді болмайды. Бұл осы қызмет үшін қосылған қызметтерді көрсетеді. Мысалы, DHCP-клиент қызметі қосалқы Winsock, Driver NetIO арналған драйвер қызметіне тәуелді болады. TDI қолдау немесе Ser-интерфейс желісі дүкені WinHTTP Auto-Discovery Service веб-прокси DHCP-клиентке тәуелді болады.

Linux қызмет конфигурациясы құралдары (11.14 суретті қар.) қызметті қарауға, іске қосуға және тоқтатуға мүмкіндік береді. Бұл құрал қызметтермен өзара әрекеттестіктің графикалық құралы түрінде болады, олар / etc/rc.d каталогтар көмегімен қадағаланады. Қызметті автоматты түрде іске қосу жүктеу барысында қызметті құру кезінде толық түсіндіріледі. Мысалы, қызметтің жүйелік журналы Syslogd және Klogd суретте көрсетілгендей іске қосылуын талап етеді. Бұл жүктеу барысында қызмет құру кезінде сипатталған. Аталған құралдың екі бетшесі бар екенін назар аудару керек, олар фондық қызметтер және талап ету бойынша қызметтер. Көптеген Linux қызметтері фон режимінде жұмыс жасауға арналған.

Linux үшін сіз кез келген сервистің қалпын, іске қосылуын қарап, оны тоқтата немесе жарлық жолағынан қайта жүктей аласыз. Жарлық жолағы қызметін басқарудың екі түрлі тәсілі бар. Қызмет көрсету жарлықтары арқылы жалпы түрде, ол / SBIN астында орналасады. / SBIN / қызмет жарлығының форматы, мұнда қызмет атауы, жүйелік журнал көрсетіледі, ал жарлық іске қосу, тоқтату немесе қайта жүктеудің бірі болады.



11.14 сурет. Linux Services сервис конфигурациясы терезесінде

Дәреже жарлығы сервис дәрежесіне жауапты болады: бағдарламаны орындау, тоқтату және бас тарту. Сервисті басқаруға арналған жарлық жолағының басқа қасиеті – бұл жарлықты тікелей сервистің өзіне беру. Мысалы, sshd сервисі /etc/init.d сақталады. Осылайша, сіз /etc/init.d/ sshd жарлығын іске қосуға (қайта жүктеу, тоқтату немесе дәреже) бере аласыз.

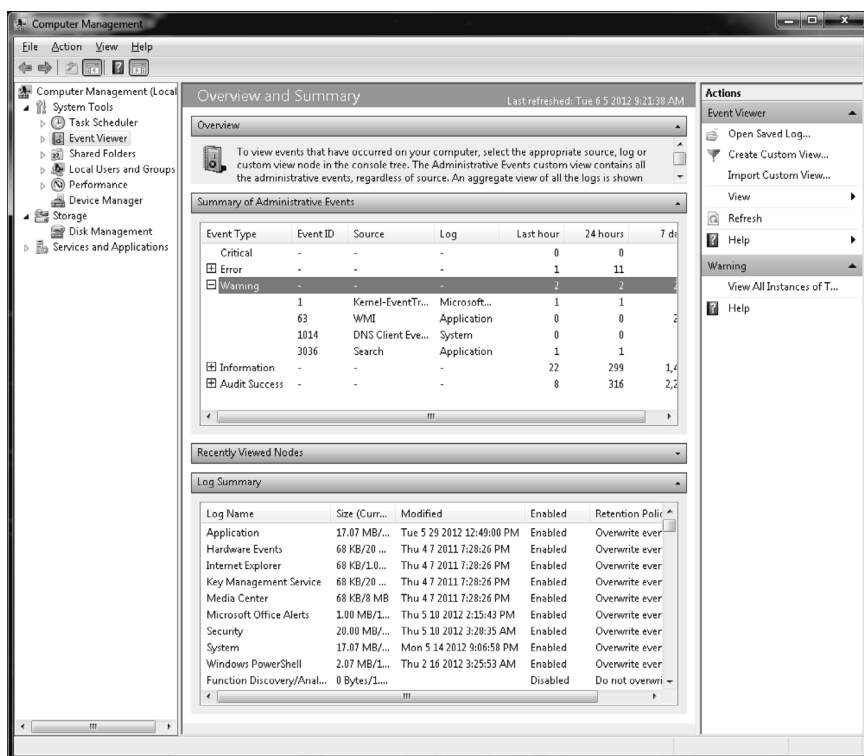
Windows және Linux сервистердің көптеген түрлері бар, олардың құқықтары да біршама өзгеше болады. Барлық сервистер жай санаттарға жіктелмейді. Сервис типтерінің жалпы тізімі қандай да бір бағдарламалармен осы сервис санаттарына қызмет көрсетеді.

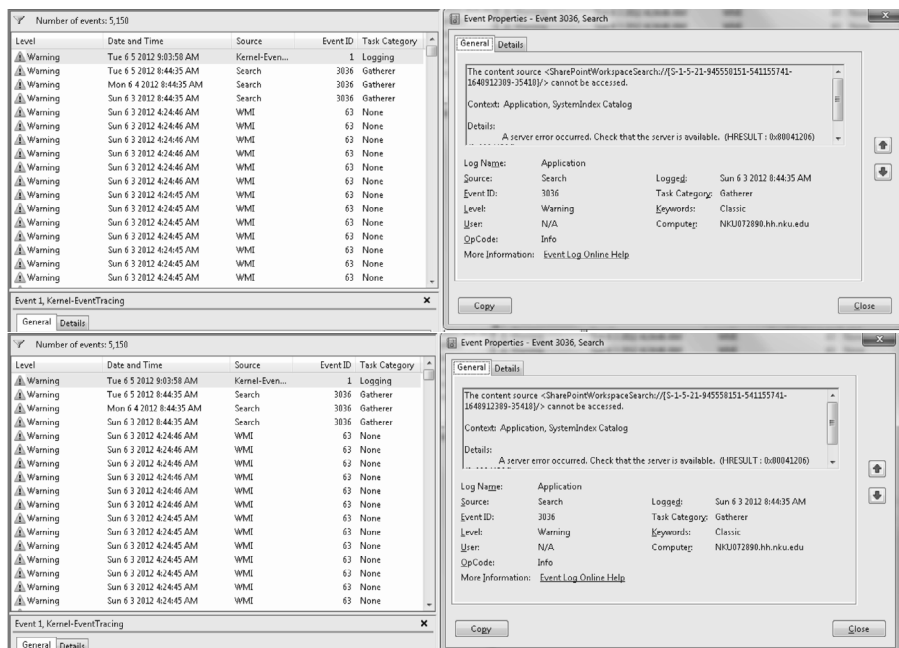
Тіркеу қызметі. Тіркеу журналдарының файлдарының ерекше немесе қате оқиға себептерін анықтаудағы маңызы зор. Windows және Linux ОЖ генерациялайтын оқиғаларды тұрақты түрде тіркейтін сервистері болады. Windows оқиғалар журналы оқиғалар туралы ақпаратты таңдауды жүзеге асырады.

Windows оқиғалардың төрт деңгейі қолданылады: аудиттен бас тарту, аудит жетістігі, ақпарат, ескерту, қателік және кризистік ескерту. Оқиғаларды қарау бағдарламасы Windows оқиғалар журналына жазылған оқиғаларды қарауға мүмкіндік береді. Журнал жазбалары қосымшалардан, Internet Explorer оқиғаларынан, Media Center оқиғала-

рынан, қауіпсіздік мәселелері оқиғаларынан және т.б. құралады. 11.15 суретте оқиғалар тіркеушісінің түрі көрсетілген (Logger events), мұнда ескертулер тізімі қамтылған, ал 11.16 суретте қосымшалардағы кейбір тіркелетін оқиғалар берілген. Linux жүйесінде Syslogd және Klogd демондары бірлесе оқиғалар есебін жүргізеді, Syslogd жүйелі қалыпты жүйедегі оқиғаларды тіркейді, ал Klogd демоны ядро генерациялайтын нақты оқиғаларды тіркейді. auditd ОЖ Linux қызметінде қауіпсіздікпен байланысты оқиғалар жинақталады және олар audit.log файлдарына жазылады. Linux – netconsole ядро модулі жүйеге кіру талпыныстарын реттейді. Басқа бағдарламалық жабдықтама өзінің жеке журнал файлдарын құра алады немесе олар syslogd хаттамасын қолданады. Linux журналдарының әр түрлі файлдары / var / linux сақталады. Мысалы, вирусқа қарсы бағдарлама жұмысты әр күн сайын түнде жоспарлауы немесе резервтік көшіру утилитасы қосылуы мүмкін.

11.15 сурет. Windows оқиғаларын қарау.





11.16 сурет. Ескерту және нақты оқиға түні бар тіркелген қосымша

Windows арналған жоспарлаушы Task Scheduler деп аталады. Ол жоспарлаушы және сервис болып табылады. Linux бірнеше жоспарлау бағдарламалары болады: at, batch, crontab. Linux жоспарлауды қолдайтын сервистері үдерістерді орындау үшін anacron болады, үдерістер өздігінен жүктеу уақытына жоспарланады, crond үдерістерді жоспарланған бағдарламамен жүзеге асырады, ал atd бағдарламалар at және batch жоспарлаған үдерістерді орындайды.

ЖАҢАРТУ ЖИНАҚТАРЫ

Жаңарту жинақтары негізінен сервистерге жатқызылмайды, алайда мұнда сервистер қарастырылып отырғандықтан жаңарту жинақтары туралы да айтып кету пайдалы болады. Жаңарту жинақтары Microsoft операциялық жүйе жаңартуларын шығару үшін қолданылады.

ОЖ NT бастап Microsoft жаңарту жинақтарын өзінің Windows ОЖ үшін шығарады. Көп жағдайда шығарылатын жаңарту жинақтары алдыңғыларын алмастырады. Мысалы 2 жинақты орнату үшін 1 жаңарту

жинағын орнату қажет емес. Алайда, бұл тәсіл барлығына бірдей жарамды бола бермейді. 3 XP жаңарту жинағы 1 жинаққа тәуелді болады. Қолданылатын жаңарту жинақтары CD-ROM жеткізіледі, алайда, қазіргі кезде оны ғаламтордан жүктеп алуға болады. Windows 7 жүйесіне орнатқан жағдайда жаңарту жинақтарын автоматты түрде жүктеп, орнатуға болады. Windows ОЖ жаңарту жинақтары шектелмеген. Басқа БЖ бөліктеріне қатысты қазір осы әдіс қолданылады.

Желілік сервистер

Компьютер желісі коммуникациясымен байланысты бірқатар міндеттер бар, сол себепті желіге қатысты көптеген желілік қызметтер пайда болған. Linux барлық желілік интерфейстерді жүктеу барысында баптайтын желің қызметтері бар. Сонымен қатар, ISDN (Integrated Services Digital Network) телефон желісіне арналған, nfsd (Network File System Daemon), nmbd (Network Message Block Daemon), ntpd (Network Time Protocol Daemon), snmptd (Simple Network Management Protocol Daemon), веб-серверге арналған сервистер (httpd), ftp-сервер (ftpd), ssh-сервер (sshd), пошталық сервер (Sendmail, Fetchmail), домендік атаулар сервисі (атауы бар). Windows операциялық жүйесінде NTDS (Network Authentication Server), BITS қызметі (Background Intelligent Transfer Service), DNSCache және DNS (DNS-сервер және клиенттер үшін), сондай-ақ, NSIS (Network Store Interface Service) ақпарат жинауға арналған белсенді желілік интерфейстері бар. Windows ОЖ жұмыс станциясы клиенттік желілік жалғауды шалғай серверлерде қолдайды. Windows ОЖ басқа желілік қызметтері Network Access Protection Agent, сонымен қатар, Network Connections и Network Locations Awareness қамтиды.

Брандмауэр және Ғаламтор қызметтері. Linux жүйесінде dnsmasq қосымшасы DNS кэш серверін қосады, ал httpd Apache веб-серверін қосу және тоқтату үшін үшін пайдаланылады. Бұл утилиталар iptables және ip6tables Linux жүйесінің брандмауэрін үнсіз келісім бойынша басқарады. Linux жүйесінің nscd сервисі құпия сөзді өңдейді және жұмыс бағдарламасын іздеу сұраныстарын топтастырады. Бұл сервис NIS және LDAP хаттамалары аутентификациясы үшін қажет. Windows ОЖ Base Filtering Engine қызметі Windows және Internet Protocol брандмауэрінің қауіпсіздігін басқарады. DHCP-клиент желілік хаттамасы IP-мекенжай және DNS-жазбаларын тіркейді және компьютер үшін жаңартады. DNS-клиент модулі DNS атауларын кэштейді. Киберқауіпсіздікті басқару ncsd секілді Windows жүйесінің Netlogon қызметі компьютер

мен домен контроллері арасындағы қауіпсіз арнаны қолдайды және ол пайдаланушылар мен қызметтер аутентификациясы үшін қажет болады.

Файлдық жүйе. Желілік қызметтер жағдайындағыдай ОЖ файлына қатысатын көптеген әр түрлі тапсырмалар бар. Linux олар екі демоннан құралады - autofs және amd, олар файл жүйесін автоматты орнату барысында жүйелік бөлімдер үшін пайдаланылады; netfs желілік қол жетімділік хаттамасы кез келген жүйе файлын монтаждайды немесе демонтаждауды жүзеге асырады. Mdmonitor файлы RAID диск массивін сақтау құрылғысының бағдарламалық бөлігін басқарады. Windows ОЖ Block level backup Engine мұрағаттау қызметі резервті көшіруді және файл жүйесінде қалпына келтіруді қолдауды қамтамасыз етеді.

Перифериялық құрылғылар. Windows ОЖ перифериялық құрылғыларды қолдауға арналған бірқатар сервистері бар.

Оларға аудио ажыратқыштар конфигурациясын қолдауға арналған Audio Service, Bluetooth қолдау қызметі және жалға және ойна технологиясы жатқызылады, ол құрылғыларды пайдаланушы деректер енгізбестен жалғауға мүмкіндік береді. Linux, CUPS (Unix жалпы баспа жүйесі) принтерлермен байланысты қолдайды. Bluetooth басқару арқылы қолдайтын бірқатар қызметтер бар, оған Bluetooth, hidd, pand және dund жатқызылады. Аферист сервисі пернетақтадан басқаруды жүзеге асырады. Gpm сервисіне мәтіндік қосымшаларға арналған тышқан жатқызылады, олар тышқан көмегімен мәтінді көшіріп қою арқылы Linux негізделеді.

Әртүрді. Linux жүйесіндегі oddjobd сервисі өзінің жеке артықшылықтары бар қосымшаларды қолдауымен ерекшеленеді. Мысалы, егер бағдарлама қандай да бір операцияларды сұрайтын болса, сондай-ақ, аталған операцияларды орындау артықшылықтары болмаса, онда бағдарлама сәйкес oddjobd сұрата алады. Windows ОЖ бірқатар диагностикалық қызметтер бар. Сонымен қатар, ата-ана бақылауын қолдау қызметтері де бар. Екі операциялық жүйе қоректі басқару қызметтерін - AMPD - Linux және Power – Windows ұсынады. Мысалы, AMPD батарея қалпын қадағалайды және автоматты түрде қолданылады.

Конфигурацияны баптау сервистері

Сервисті іске қосқанда ол негізінен конфигурация файлы ретінде оқылады. Конфигурация файлы сервис орындалатын қызметтерді анықтайтын болады. Мысалы, Linux брандмауэрі сервисі қандай да бір ережелерді пайдаланып жұмыс жасайды. Негізінен ережелер конфигурация

файлында белгіленеді. Windows сервистері ini (инициализация) кеңейтілуімен жұмыс жасайды, ал кейбір жағдайда олар .cfg немесе басқа кеңейтілу түрінде немесе кеңейтусіз болады. Linux жүйесінде конфигурация файлдары .conf кеңейтілу түрінде болады, дегенмен көп жағдайда бұл орын ала бермейді. Linux жүйесінде файл жолағын бастау үшін # таңбасын қолдануға болады.

Мұнда біз Linux.conf бірнеше файлдарына баса назар аударамыз, олардың синтаксисін зерттеп, кейбір өзгерістіерін қарастырамыз. Linux жүйесінде конфигурация файлдарының көпшілігі директориядан табылады.

Біз қызметтердің бірі - Linux – Syslogd жүйесін тіркеуден бастаймыз. Бұл сервис қосымшаның БЖ генерациялайтын оқиғаларды тіркейді. Syslogd конфигурациясы үшін келесілерді білу маңызды:

1. Қандай қызмет түрлері тіркелуі керек
2. Қандай шарттар журнал жүргізуді талап етеді
3. Тіркеу қайда орын алуы керек

Файл /etc/syslog.conf осы ақпаратты сипаттайды. Аталған файлдың әр жолағы оқиғалар журналын жүргізудің бір формасын сипаттайды. Хаттамалар оқиғалары бір немесе бірнеше көздерден құралады (бағдарламалық жабдықтама), олардың басымдылық деңгейі оқиғаны тіркеу себебі болады.

Кез келген жеке жазба форматы source.priority [; source.priority] * амалы болып табылады. Әрбір жазба кем дегенде source.priority әрекеті түрінде болады. Топтастыру таңбасын (*) пайдалану синтаксисте «0 немесе одан артық» көшірмелерді білдіреді. Кез келген дерек көзі үшін немесе басымдылық үшін пайдаланған жағдайда * «кез келген дерек көзі» немесе «қандай да бір басымдылық деңгейін» білдіреді. Мысалы, авторландыру хабарламасын көрсету үшін auth. * пайдалануға болады. Не болмаса кез келген басымдылық деңгейі апатты хабарламаны көрсету үшін * .emerg жазады. *. * жазбасы кез келген оқиғаны БЖ көмегімен тіркейді.

Файлдың орналасқан орнын көрсетуден басқа * таңбасын барлық тіркелген пайдаланушыларға хабарламаны жөнелту үшін терминалдың ашық терезесінде пайдалануға болады. Басқа балама әрекет – бұл хабарламаны тіркеудегі тік сызықша, ол БЖ басқа бөлімінде болады. Мұндай жазба | Exec / /usr/local/bin/filter түрінде болады.

Үнсіз келісім бойынша бірнеше дерек көздері бар. Оларға auth (пай-

даланушы аутентификациясы сервисі), cron (cronjob жоспарлаушы), daemon (барлық стандартты қызметтер), kern (Linux жүйесінің ядросы), lpr (принтер серсері), mail (пошталық сервер), syslog (syslogd сервисі), және пайдаланушы бағдарламалары (пайдаланушы қосқан бағдарламалар) жатады. Пайдаланушы бағдарламалары local0-local7 түрінде белгіленеді, бұл сегіз қосымша бағдарламамен жұмыс жасауға мүмкіндік береді, олар журнал жүргізу үшін бапталады. Тоғыз деңгей 11.2 кестеде берілген.

12.1 кесте. syslog.conf пайдаланатын басымдылықтың тоғыз деңгейі

None ешбір басымдылық жоқ

Debug реттеу хабарламасы, бағдарламалаушылар өз бағдарламаларын сынау кезінде пайдаланады

Бағдарламаны орындау туралы ақпараттық хабарлама

Назар аударыңыз. Назар аударатын оқиғалар жоқ

Ескерту. Мүмкін мәселелер туралы ескерту

Қателік. Орындау барысында туындайтын қателіктер.

Крит Хабарлама, кризистік хабарламаларды сипаттайды, олар бағдарламаны апатты аяқтау нәтижесі болады.

Ескертпе хабарламаларда бағдарламаны апатты аяқтауға алып келетін қателіктер сипатталады және олар басқа бағдарламаларға әсер етеді.

Апаттық хабарламалар жүйе жұмысының тоқтауына алып келетін жағдайлар туралы ескертеді.

11.17 суретте syslog.conf файлының мазмұны көрсетілген. Алғашқы жазба ядродағы кез келген хабарламаның / Dev / консоль жіберілгенін көрсетеді, ол негізгі компьютер консолі болып табылады (бұл хабарлама жүйеге қашықтықтан немесе басқа терезеден кіру консолі болады). Екінші жазба көптеген хабарламаларды анықтайды: барлық ақпараттық хабарламалар және барлық пошта, authpriv және cron қандай да бір басымдылығы жоқ хабарламалар. Олардың балығы хабарламалар / Var / Журнал / журналына енгізілген. Үшінші жазба барлық басқа хабарламалар authpriv (басымдылық деңгейі жоғары – «басымдылығы жоқ») қауіпсіздік журналына / Var / Журнал / жазылғанын көрсетеді. Басқа жазбалардың мағынасын түсіне аласыз ба? Файл атауының алдында "-" пайдалану - / Var / Журнал / MAILLOG суретте хабарламаның синхрондалмайтынын көрсетеді.

Linux брандмауэрі конфигурация файлының болуын талап етеді. Бұл файл / etc / sysconfig / ipt сақталады. Файл брандмауэр ережелерін қосуға, өзгертуге және жоюға мүмкіндік береді.

11.17 сурет. /etc/syslog.conf тіркеуіне ену

```
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                     /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none    /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                    /var/log/secure

# Log all the mail messages in one place.
mail.*                                        -/var/log/maillog

# Log cron stuff
cron.*                                       /var/log/cron

# Everybody gets emergency messages
*.emerg                                     *

# Save news errors of level crit and higher in a special file.
uucp,news.crit                             /var/log/spooler

# Save boot messages also to boot.log
local7.*                                    /var/log/boot.log
```

Ережедегі келесі тармақ бұл хабарлама келетін ағын атауы. Ағындардың INPUT, OUTPUT немесе FORWARD сияқты жалпы атаулары болады немесе белгілі бір жерден болуы мүмкін, мысалы, fast – input-queue немесе icmp-queue - out. Төменде келтірілген мысалдар input stream RH-Firewall-1-INPUT көрсетеді.

Одан әрі ереже сәйкес келетін нақты жағдайлар жүреді. Олар порттардан, TCP / IP хабарлама түрлерінен, көзінен немесе IP – тағайындау мекенжайынан немесе протоколдардан тұруы мүмкін. Мысалы, --dport 50 50 порты арқылы хабарламаны білдіреді, ал -p udp --dport 5353 5353 тағайындалу udp-порт арқылы хабарламаны білдіреді, а -d 1.2.3.4 1.2.3.4. IP – тағайындау мекенжайын білдіреді.

Ережедегі соңғы жазба – бұл егер бұл ереже хабарламаға сәйкес келетін болса, ереже қолданылатын қарапайым әрекет. Ең негізгі әрекеттері ACCEPT, LOG, DROP және REJECT. ACCEPT хабарламаны өңдеуді тез арада тоқтатады және оны ОЖ жібереді (яғни брандмауэрдің жанынан). LOG әрекеті хабарламаны тіркейді, бірақ егер басқа ережелер болашақ әрекеттерді нұсқау үшін жарайтын болса, онда ереже негізінде өңдеуін жалғастырады. DROP хабарламаны блоктауға алып келеді, ал REJECT

хабарламаны блокуйды, бірақ сонымен қатар пакет блоктанған басты компьютерге пакетті жөнелтуге жауап береді.

Төменде ережелердің бірнеше мысалдары келтірілген:

```
-A RH-Firewall-1-INPUT --dport 50 -j ACCEPT
```

```
-A RH-Firewall-1-INPUT -p UDP --dport 5353 -d 224.0.0.251 -j ACCEPT
```

```
-A RH-Firewall-1-INPUT -p TCP -m TCP --dport 631 -j ACCEPT
```

Негізінен, Iptables файлы әдепкі қалпы бойынша ережемен аякталады. Егер алдыңғы ережелер қабылдау және кіру үшін хабарламаны анықтайтын болса, онда келесі ереже қалған барлық ережелерді қабылдамау үшін «backstop» болады.

```
-A RH-Firewall-1-INPUT -j REJECT --reject-icmp-host-prohibited
```

/ etc / fstab файлында орнату туралы ақпарат сақталады. Бұл орнату (барлығын орнату) командасы шықпаған әр жағдайда қолданылады. Конфигурация туралы ақпарат орнату, орнату нүктесі (каталог) бойынша әрбір тарауда айтылып шығады, сонымен қатар осы тарау туралы нақты ақпарат, мысалы тараудың жеке жазбалар үшін бөлінген дисктік кеңістігі бар ма немесе тарау тек оқу үшін ғана қолжетімді ма екені хабарланады.

LABEL = / Home / ext3 әдепкі қалпы бойынша 1 2

Бұл бапта жеке тараудың директорияды/жеке бөлімде орналасуы қажет екені сипатталады, бөлім типі ext3 болып табылады, бөлім үнсіз келісім бойынша белгіленген параметрлерді пайдаланады, олардың мәні сәйкесінше 1 және 2 болады, жүктеу жиілігі (мұрағаттау кестесі) үшін арналады және номерді береді (тапсырысты қадағалайды, ал fsck бағдарламасы бөлімдерді тексереді). Опциялар ro (тек оқуға ғана арналған), rw (оқу-жазу), auto немесе noauto (жүктеу барысында жаңа бөлім автоматты түрде орнатылуы керек пе немесе керек емес пе), сонымен қатар, usrquota (пайдаланушының әр каталогында белгіленген көлемді анықтау) қамтиды. Опциялар ext3 типінен құралуы мүмкін және олар Linux жүйелерінде жиі кездеседі. Дегенмен де, бұл бөлім желі бойынша (nfs) файлдық жүйе арқылы орнатылған, IP- мекенжайының қосымша параметрі addr көмегімен #. #. # тең болады, мұндағы # таңбалары IP мекенжай октеттері болып табылады.

/etc/resolv.conf конфигурация файлы ең қарапайым конфигурациялық файлдардың бірі болып табылады. Ол IP-мекенжайларды жүйенің доменді атауларында (DNS-серверлер) компьютер үшін сақтайды. Ал-

дыңғы екі конфигурация файлдарының мысалдарынан айырмашылығы-
бұл файл жазбалардан құралады және келесі түрде болады.

10.11.12.13 атаулар сервері

10.14.15.16 атаулар сервері

resolv.conf файлын Linux көптеген бағдарламалары пайдаланады. DNS 12 тарауда қарастырамыз. Linux пошталық сервері бұл sendmail. Оның бірқатар конфигурация файлдары бар. Солардың бірі / etc./ aliases электронды поштаға орналастырылатын лақап аттар береді. Мысалы, bin, daemon, adm секілді мекенжайлар мен пошта түбегейлі каталогта лақап аттармен қамтамасыз етілген, сол себепті хабарламаса осы орындардың кез келген біріне жіберілген жағдайда, ол шын мәнінде түбегейлі каталогқа жіберілген болады. Келесі конфигурация файлы бұл - / etc. / mail / sendmail. cf. Дегенмен, бұл файлды ешқашан тікелей редакциялай алмайтыныңыз қарастырылған. Соның орнына / etc. / mail/ sendmail. mc файлы макро файл болып табылады. Оны редакциялау сізге жаңа sendmail.cf файлын құруға мүмкіндік береді. Sendmail аса күрделі бағдарлама болғандықтан, біз бұл конфигурация файлын қарастырмаймыз.

Бір соңғы конфигурация файлы ретінде ldap.conf айтып кетуге болады. LDAP сервер болып табылады және қол жетімділік құқықтарын желілік тексеру үшін қолданылады. Яғни, компьютерлер желісін ескере отырып, LDAP серверін пайдаланушыны желілік компьютердің кез келген бірінде аутентификациялау үшін пайдалануға болады. LDAP конфигурация файлының форматы алдыңғы файлдардан өзгеше болады, мұнда әрбір жазба бағдарламаға нұсқаулық немесе ереже түрінде болады. Оның орнына LDAP конфигурация файлында директивалар болады. Директивалардың формалары өте көп, олардың әр қайсысы LDAP серверін дайындау бойынша әр түрлі міндеттерді атқарады. Мысалы, базалық директива іздеуге арналған «ерекше» атауды көрсетеді. Басқа игі директивасы домен сокеттеріне арналған IP – мекенжайын жариялайды. Басқа директивалар уақыт шектеулерін, порттар, филтрлер, атау контексттерін, карта атауларын және т.б. анықтайды. Төменде ldap.conf файлының мысалы берілген:

URI LDAP: //127.0.0.1/ ssl no

tls_cacertdir / etc / open ldap / cacerts pam_ password md5

Сервистің конфигурация файлын өзгерту автоматты түрде сервистің

осы өзгерістерді қабылдауының себебі болмайды. Көп жағдайда сервис қайта қосылуы керек. Сіздер сервисті тоқтату алдында және оны қайта іске қосу алдында жүзеге асыра аласыздар. Көп жағдайда конфигурация файлына өзгеріс енгізгенге дейін сервисті тоқтату маңызды болады.

ЖҮКТЕУ БАРЫСЫНДА СЕРВИСТІ АНЫҚТАУ

Біз осы тарауды жүйені жүктеу немесе баптау барысында қандай сервистер автоматты түрде іске қосылатынын және қандай сервистер қосылмайтынын қарастырумен аяқтаймыз. Жоғарыда айтып кеткендей, Windows сервистерін Services құралдары көмегімен басқаруға болады. Осы құралмен кез келген қызметті таңдау арқылы сіз сервисі іске қосып немесе тоқтата аласыз. Сервис қасиеттері терезесі арқылы іске қосу типін де көрсетуге болады. Келесі нұсқаларды жүзеге асыру мүмкіндігі бар – автоматты (кейінге қалдырылған іске қосу), автоматты қолмен, іске қосуды сөндіру. Администратор ретінде сіз кез келген немесе барлық сервистердің іске қосылу типін өзгертіп немесе өз қалауыңызға сәйкес түрлендіре аласыз. Мысалы, ол ата-аналық қадағалау үнемі жүктеу барысында қосылуы қажет деп есептейтін болса, сіз іске қосу типін қолмен қосудан автоматты түрге ауыстыра аласыз. Бұдан басқа, Bluetooth қолдау қызметін де қосу мүмкіндігі болады.

Linux жүйені баптау барысында белгілі сервистерді автоматты іске қосуға бапталған, сонымен қатар, сервистерді жүйелік администраторлар реттей алады. Дегенмен де, Windows айырмашылығы Linux әр түрлі іске қосу сервистерін орындалу тәсіліне тәуелді орната алады. Linux Run жеті деңгейі болады. Олар 11.3 кестеде берілген.

Linux алғаш жүктеліп жатқанда ол `script / etc / inittab` іске қосады. Inittab орнататын алғашқы заттардың бірі жұмысты іске қосу деңгейі болып табылады. мысалы, 5 іске қосу деңгейі операциялық жүйені толық экранды режимде іске қосады және оның графикалық интерфейсі болады, ал 3 іске қосу деңгейі толық мәтінді режим түрінде болады, 1 іске қосу деңгейі бірегей пайдалану мәтіні режимі болады. Іске қосу деңгейіне байланысты қызметтердің жұмысты іске қосу деңгейі негізінде қосылғаны анықталады. Inittab скрипт `script/etc/rc.d/rc` орындайды. Бұл скрипт іске қосу барысындағы бірнеше амалдарды қадағалайды, алайда, олардың арасында `/etc/rc.d/rc#.d` каталогын таңдайды, мұнда # іске қосу деңгейі болып табылады (мысалы, 5 жұмыс деңгейі үшін `rc5.d`).

`/etc/rc.d/rc5.d` қарау барысында бірқатар символдық сілтемелер көруге болады. Әрбір символдық сілтеменің `K ##` немесе `S ##` түріндегі атаулары болады. Мұндағы K және S түріндегі сілтеме атаулары сәйкесінше

«жою» мен «іске қосуды» білдіреді. Бұл хат RC сценарийі сервисті іске қосу барысында жою немесе қосу мүмкіндігін білдіреді. Саны сервис қосатын (немесе тоқтатылатын) ретін білдіреді. 11.8 суретте 5 орындау деңгейіндегі барлық символдық сілтемелер берілген. /etc/rc.d/rc5.d және /etc/rc.d/rc3.d каталогтары арасындағы салыстыруды графиктік интер-фейске қажетті сервистер көрсетеді, олар желідегі коммуникация үшін қажет болып табылады.

Бұл каталогтағы әрбір элемент тек символдық сілтеме болғандықтан, rc script сілтеме бойынша қолданыстағы сервиске ауысады. Барлық сер-вистер /etc/rc.d/init.d каталогында сақталады, 11.9 сурет.

11.3 кесте. Linux ОЖ іске қосу деңгейлері

Сипаттама

0 Тоқтату жүйе қайта жүктелмеген жағдайда барлық қызметтерді сөн-діреді

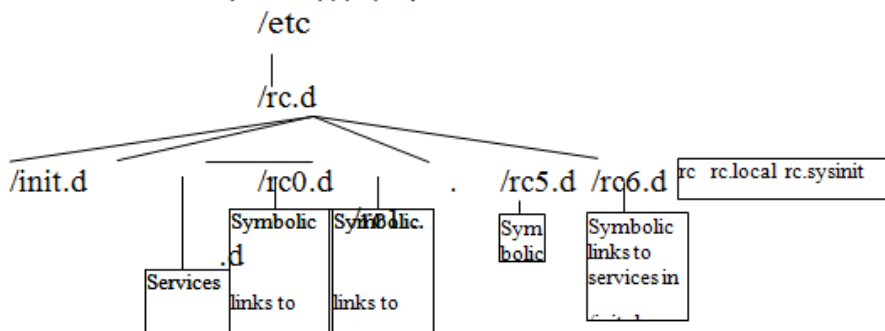
1 Бір пайдаланушы режимі, жүйеге қызмет көрсетуге арналған, желілік мүмкіндіктерсіз түбегейлі каталог ретінде қызмет етеді

2 Көп пайдаланушы режимі бірінші кезекте техникалық қызмет көрсету және тестілеу үшін пайдаланылады

K01dnsmasq	K50snmpd	S07iscsid	S19rpcgssd	S80sendmail
K02avahi-dnscnf	K50snmptrapd	S08ip6tables	S22messagebus	S85gpm
K02NetworkManager	K69rpcsvcgssd	S08iptables	S25bluetooth	S90crond
K02odddj	K73ypbind	S08mcstrans	S25netfs	S90xfs
K05conman	K74nscd	S09isdn	S25pcscd	S95anacron
K05saslauthd	K74ntpd	S10network	S26acpid	S95atd
K05wdaemon	K85mdm	S11auditd	S26apmd	S96readahead_early
K10psacct	K07multipathd	S12restorecond	S26haldaemon	S97yum-updatesd
K10tcsd	K88wpa_supplicant	S12syslog	S26hidd	S98avahi-daemon
K15httpd	K89dum	S13cpuspeed	S26lm_sensors	S99firstboot
K20nfs	K89netplugd	S13irqbalance	S26lvm2-monitor	S99local
K24irda	K89pand	S13iscsi	S28autofs	S99smartd
K35vncserver	K89rdisc	S13portmap	S50hplip	
K35winbind	K91capi	S14nfslock	S55sshd	
K50ibmasm	S04readahead_early	S15mdmonitor	S56cups	
K50netconsole	S05kudzu	S18rpcidmapd	S56rawdevices	

11.18 сурет.

/etc/rc.d/rc5.d сервистерді қосу каталогы



11.19 сурет. /etc/rc.d. каталогының құрылымы. Іске қосу сценарийлеріне арналған директория асты құрылымын көркемдейді. Сонымен қатар, /etc/rc.d каталогында екі қосымша скрипт бар: rc.sysinit және rc.local. rc.sysinit скрипт бірқатар жүйелік баптауларды, пайдалану үшін батырмаларды, қоршаған орта айнымалыларын жүктейді, сонымен қатар, құрылғыны баптау бойынша басқа әрекеттерді орындайды. Rc.local script ортаны одан әрі баптау үшін жүйелік әкімгер үшін қолжетімді. Ол скриптті тек бір рет орындайды, жүктеудің барлық басқа скриптері өздері орындалады. Бұл файл бастапқыда одан бастау үшін бос дерлік. Жүйелік әкімгер ретінде сіз символдық сілтеме аттарын қарапайым өзгертуді орындау деңгейін қосымша бейімдей аласыз. Мысалы, егер ол жүйелік әкімгер болып саналса, онда Bluetooth қызметі 5 деңгейде қажет болмайды, жүйелік әкімгер /etc/rc.d/rc5.d S25bluetooth символдық сілтемесін K25bluetooth атын өзгерте алады. Яғни, Bluetooth сервисін іске қосудың орнына ол жойылады. Әрине, егер сіз мұны білмесеңіз, онда бұл символдық сілтемелерді жайына қалдырған жөн. Сіз бастау/тоқтату жүйесі сервисінің /SBIN Bluetooth пайдалана отырып, айтарлықтай соңғы кезеңінде сервистерді іске қоса немесе тоқтата аласыз.

12. Үдерістер мен қызметтер

Бұл тарауда компьютерлік желілер сипатталады. Бұл тарау компьютерлік желінің физикалық құрамдауыштарын құрайтын компьютерлік құрылғыны сипаттаудан басталады. Бұдан әрі, желілер архитектура деңгейінде талқыланады: желілер мен желілік хаттамалар жіктелімі. Желілік бағдарламалық жабдықтама, оның ішінде, Linux арнайы бағдарламалық жабдықтама тарауға енгізілген. Және соңында, аталмыш тарауда Ғаламтор қарастырылады: ол қалай жұмыс жасайды, ол қалай пайда болды және оның болашағы қандай болады.

Аталмыш тарауды оқыту мақсаттары бұл:

- Тарату желісі құрылғыларының рөлін сипаттаңыз.
- Желілік орта типтерін ажыратыңыз.
- Желілік топология мен жіктелімдерін салыстырыңыз.
- TCP / IP хаттама жігінде және OSI моделінде әрбір деңгейдің рөлін талқылаңыз.
- IPv4 дербестендіру әдістерін түсіндіріңіз және оны IPv6 салыстырыңыз.
- Желілік бағдарламалық жабдықтаманың танымал формаларын енгізіңіз.
- Ғаламтордың қысқаша тарихын беріңіз.
- Ғаламтор арқылы деректер алмасуы қалай жүретінін сипаттаңыз.

Компьютерлік желі компьютерлер деректер мен ресурстармен алмаса алатындай түрде қосылған компьютер және компьютерлік ресурстар жиынын (мысалы, принтерлер, файлдық серверлер) көрсетеді. Компьютерлік желілер арқылы адамдар деректер мен ресурстармен алмаса алады. Компьютерлік желілер арқылы адамдар сөйлесе, ақпарат алмаса алады, құрылғыны бірге пайдалана, деректерді оқшаулай және қорғай алады, сонымен қатар деректерді қолайлы сақтық көшіру үшін платформаны қамтамасыз етеді. Желілер (оның ішінде, Ғаламтор) сондай-ақ, са-

уда мен сату үшін құралдар ұсынады. Шынында да, Ғаламтор ұсынған бірқатар түрлі артықшылықтар бар.

Дегенмен желілер деректерді, ресурстарды және байланысты бірігіп пайдалану жолымен жұмыс орнындағы тиімділікті арттыру үшін таңдалған, қандай да бір компьютерлік желімен байланысты бірқатар шығындар бар. Физикалық тасымалдағыштарды алу мен орнату және желіні әкімшілендіру үшін құны бар. Алайда бұл мәселе емес, себебі желіні қорғау құны мен қорғалмаған желінің болу құны біршама ерекшеленеді. Бұдан басқа, компьютерлік желілер көмегімен көптеген адамдар Ғаламтор қолжетімділігіне ие болады, сонымен қатар жұмыс орнында бұл лайықсыз тәртіп пен уақытты тиімсіз пайдалануға алып келуі мүмкін. Компьютерлік желіні қамтамасыз етуге қажетті компьютерлік желі мен шаралардың қатерлерінің кейбіреулері 15 тарауда қарастырылған.

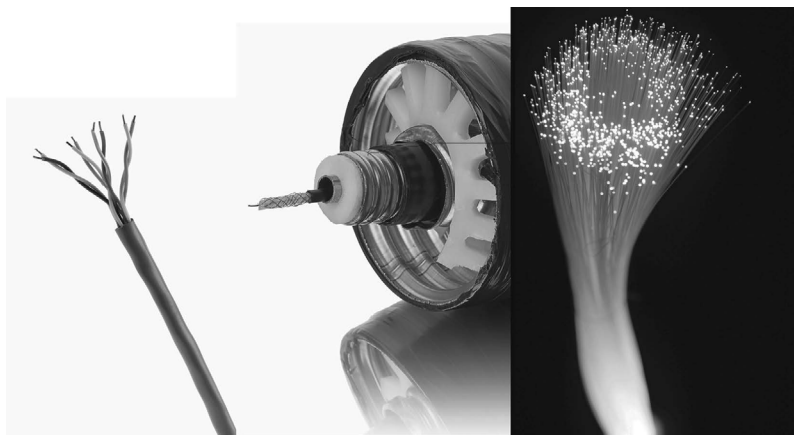
Дегенмен де желі ресурстардың көптеген түрлерінен құралуы мүмкін, бұл компьютер желі бойынша сөйлесу үшін пайдаланылатын негізгі құралы болып табылады. Желіде біз компьютерді жергілікті немесе қашықтағы сияқты анықтаймыз. Жергілікті компьютер бұл қолданушы пайдаланатын компьютер. Яғни, қолданушы физикалық түрде осы компьютердің маңайында болады. Қашықтағы компьютер желі бойынша қолжетімділікті жүзеге асыруға арналған компьютерді береді. Мысалы, қашықтағы жұмыс үстеліне немесе telnet [немесе ssh (secure shell)] қосылу арқылы сіз басқа компьютерге кіре аласыз. Компьютерге кірерде қашықтағы компьютер екені ескертіледі. Хост қашықтағы тұрғылықты жерде тіркелген болуы мүмкін компьютер типін көрсетеді. Бұрындары бұл маңызды еді, себебі дербес компьютерлер хост-компьютер бола алмайтын. Алайда қазіргі таңда көптеген компьютерлер қашықтағы кірістің кейбір түрлеріне немесе қашықтағы жұмыс үстеліне қосылу мүмкіндік бергендіктен тіптен қажет емес.

Компьютерлік желілерді физикалық деңгейде (байланыс мүмкін болатын қосылыстар немесе физикалық тасымалдағыштар), логикалық деңгейде (желілік топология) немесе бағдарламалық деңгейде (компьютерлерге байланысуға мүмкіндік беретін хаттамалар мен бағдарламалар) қарастыруға болады. Осы тарауда бұл деңгейлердің әрқайсысындағы компьютерлік желілер негізінде жатқан идеялардың кейбіреулері қарастырылады.

ФИЗИКАЛЫҚ ДЕНГЕЙДЕГІ ЖҰМЫС ЖЕЛІЛЕРІ

Желінің физикалық деңгейі желі бойынша ақпарат алмасу қалай жүзеге асырылатынын анықтайды. Осы деңгейде ақпарат медиа-физикалық

байланыстың кейбір типтерінің ішінде бит түрінде беріледі. Бұқаралық ақпарат құралдары ақпаратты электр тоғы, электромагниттік толқын, жарық импульстері немесе радиотолқын түрінде береді (кейде қысқа толқынды пештер сияқты аса жоғары жиілік кезінде). Беру формасы таңдалған тасымалдағыш типіне негізделеді. Тасымалдағыш типі кем дегенде желідегі ресурстар қашықтықтар негізінде таңдалған. БАҚ формасы желінің жіберу қабілетін айтады. БАҚ арналған қолжетімді мәліметтерді берудің бұл жылдамдығы жіберу қатарының ені секундтағы биттің кейбір саны ретінде сипатталады (бит немесе б/с). Жіберудің заманауи жолақтары секундына миллион бит (Мбит / сек, Mbps). Компьютерлік модем сияқты неғұрлым ескі технологиялар (төменде сипатталған) секундына жүз немесе мыңдаған битке дейін шектелген болатын, мысалы 56 Кбит. Қазіргі таңда пайдаланылып жүрген желілік қосылыстың аса кең таралған формасы коаксиальді шоғырсым және талшықты-оптикалық шоғырсым. Бұрындары аса кең таралған байланыс ол Құрама Штаттарында телефон желісінің көп бөлігінде қолданғандай, сымдардың тарамдалған жұбы бойынша болды. Біз әлі күнге дейін тарамдалған жұпты пайдаланамыз, себебі бұл арзан және оның көптеген бөлігі салынған. Коаксиальді шоғырсым да және тарамдалған жұп та ақпаратты электромагниттік толқын көмегімен береді, бұл уақытта талшықты-оптикалық шоғырсым жарық импульстерін қолданады. Үлкен қашықтықтар үшін шоғырсым тым қымбат және сол себепті радиосигналдар радио мұнаралар, ұялы байланыс мұнарасы, қысқа толқынды мұнара бойынша беріледі және орбитада спутниктерден әсер береді.

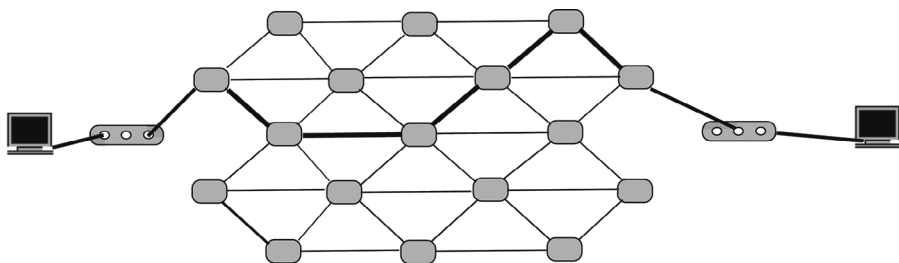


12.1 сурет Шоғырсым түрлері.

Сымдардың тарамдалған жұптарын (осы суретте төртеуі біреуіндегі шоғырсым), коаксиальді шоғырсым, сонымен қатар талшықты-оптикалық шоғырсымның ондаған тарамдарын салыстыру үшін 12.1 суретті қараңыз.

Осы физикалық деңгейде желі сигналдарға деректерді кодтау мен қайта кодтауға, сигналдарды модуляция мен демодуляциясына, сигналдарды беру мен қабылдауға және сигналдарды бағыттауға жауап береді. Жіберу желі мұны хабарламаны жіберу кезінде орындайтын соңғы қадам, ал қабылдау хабарламаны алу кезіндегі алғашқы қадам болып табылады. Кодтау/қайта кодтау компьютерде сақталған формада желі талап ететін формаға хабарламадағы жеке биттердің ауысуын талап етеді. Компьютер жадында сақталатын хабарламалар электр зарядтарынан (тоқтарынан) тұрады, оптикалық-талшықты шоғырсым бойынша жіберу кезінде хабарлама жарық импульстерінің сериялары болады. Егер физикалық тасымалдағыш сандық сигнал емес, аналогтік сигналды тасымалдаса, одан кейінгі тасымалдауы қажет, ол модуляция ретінде айқын. Сигнал телефон желісі бойынша берілгенде 1 және 0 жүйелілігі таратылатын сарынға ауысады. Демодуляция аналогтік сигналдан бастапқы сандық сигналға ауыстырады. Ақырында бағыттауыш желінің бір жерінен келесі жеріне хабарламаны жібереді.

Бағыттау формасы желілік пакет қосылғанына немесе каналдар коммутациясы орындалғанына тәуелді. Каналдар коммутациясы бар желілер толық жол немесе тізбек жіберу барлық ұзақтық бойы басталатынына және қолдауына дейін орнатылуы керек болуы үшін талап етіледі. Телефон желісі каналдар коммутациясы бар желіні көрсетеді. Сондықтан да бағыт бар, онда бүкіл сөйлесу бойы кез келген бағыттағы байланыс бір уақытта жүруі керек. Коммутация пакеті бар желілерде хабарлама бағыты енді орнатылады, себебі ол жіберіледі. Хабарлама бір жерде алынса және егер ол тағайындау орны болмаса, онда хабарлама басқа жерге жолданған. Жолды таңдау желі қолжетімділігі мен хабарлама трафигі көлеміне негізделген. Ғаламторы бар, коммутация пакеті бар көптеген компьютерлік желілер айтарлықтай танымал және жиі мысал келтірілетін болып табылады. 12.2 суретте хабарлама тағайындау пунктіне жеткенше бір орыннан екінші орынға жіберілетін желі көрсетілген. Егер коммутация каналы бар желі бар болса, онда бағыт ертерек құрылады. Егер коммутация пакеті бар желі болса, бағыт бір реттегі бір тармақ сияқты құрылады. Осылайша, пакеттік коммутацияда екі ресурс арасындағы бірнеше хабарламалар түрлі жолдарды пайдаланып, желіні қиып өтуі мүмкін.



12.2 сурет Желі арқылы жүру.

Желілік ресурстарды бірге байланыстыратын физикалық ортадан басқа, хабарламаны бір жерден екінші жерге жіберу үшін қолданылатын басқа да құрылғылар бар. Тарату құрылғылары шоғырлауыштардан, коммутаторлардан, бағыттауыштардан және шлюздерден тұрады. Барлығы бірге бұл құрылғылар нүктелерді беру арқылы желілерді бірге «жабыстырады», мұнда желі құрылғыларға және басқа желілерге қосыла алады.

Желілік шоғырлауыш бұл бірнеше компьютерлерді бірге жалғайтын қарапайым құрылғы. Шоғырлауышқа жалғанған бірнеше компьютерлер болған кезде шоғырлауыш қабылдаған хабарлама желідегі барлық компьютерлер үшін қолжетімді болады. Хабарламаға бекітілген мақсат мекенжайы хабарлама қай компьютерге тағайындалғанын көрсетеді, бірақ бұл басқа компьютерлерге хабарламаны таңдауға кедергі келтірмейді. Желілік коммутатор шоғырлауышқа қарағанда айтарлықтай қабілетті болып табылады. Бір жағынан, коммутаторға жалғанған барлық компьютерлердің жергілікті желілік мекенжайларын [(MAC) мекенжай ортасындағы қолжетімділікті басқару] жазады. Хабарлама содан кейін тағайындау мекенжайына сәйкес келетін компьютермен бірге жіберіледі. Осылайша, коммутатор желінің жіберу қабілетін айтарлықтай тиімді пайдаланады.

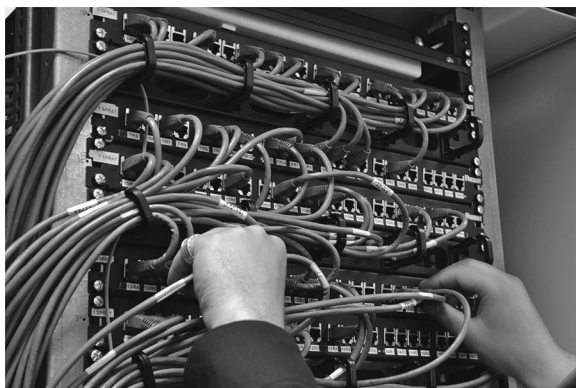
Бағыттауыш бірнеше желілерді бірге жалғайтын құрылғыны көрсетеді. Осылайша, бағыттауыш ауыстырып-қосқыш үшін ауыстырып-қосқыш сияқты. Дегенмен сіз бағыттауышты компьютерлерді тікелей желіге қосу үшін пайдалана аласыз, бағыттауыштар әдетте хабарлама басқа желілерге жіберілу үшін желінің нақты орындарында пайдаланылады. 12.3 сурет желілік шоғырлауышты береді және 12.4 сурет қосылған құрылғылары бар бірнеше желілік коммутаторларды көрсетеді. Сырттай бағдарлауышы бар коммутатордан шоғырлауышты саралау үшін аздаған айырмашылық бар (өлшемінен басқа). Іштей коммутаторда көптеген құрылғы, оның ішінде шоғырлауышқа қарағанда, MAC-ме-

кенжайларды сақтау үшін орын да бар. Бағдарлауышта бағыттаудың бағдарламалық кестелері бар және оған басқа желіден келетін ең кем дегенде бір кіріс кіреді.



12.3 сурет Желілік шоғырлауыш

Желі, БҚ және ғаламтор



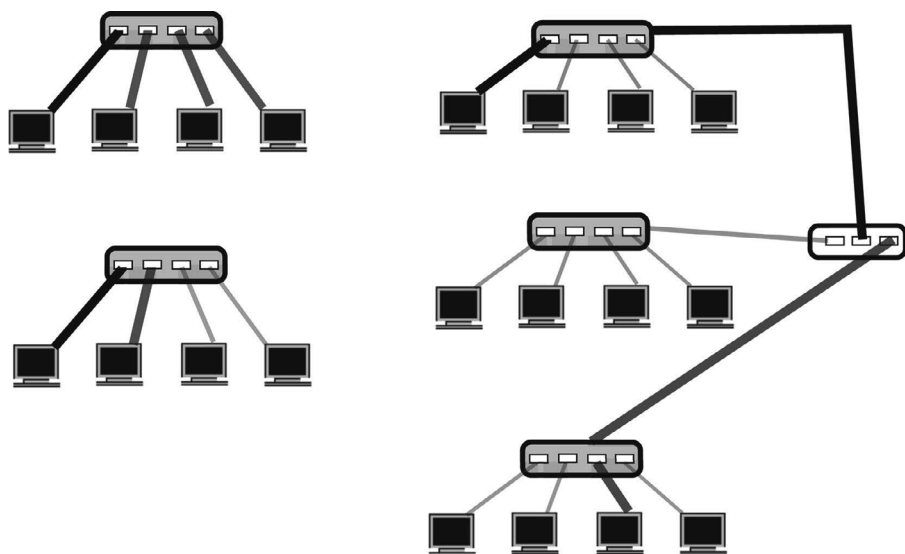
12.4 сурет Желілік коммутатор

Коммутатор және бағдарлауыш кірген хабарламаны бағыттау үшін желіні таңдауға хабарламада алушының мекенжайын пайдаланады. Бағдарлауыш шешімдер қабылдау үшін қосымша мүмкіндіктерге ие. Мысалы, егер трафик хабарламасы оның сыйымдылығынан асса, бағдарлауыш хабарламаны мақсатты түсіруге тура келеді. Бұдан басқа, бірнеше кіріс хабарламалары болған кезде бағдарлауыш қайсысын бірінші жіберу керек екенін таңдау керек. Бағыттау шешімі бұл хабарлама орналасқан тағайындау желісі. Бұл шешімді ережеге сай, бағыттауыш жадында сақталатын бағыттау кестесіне дейін қалдырады. Желілік шлюз түрлі типті желілерді жалғайтын бағыттауышты береді. Яғни, егер түрліше хаттамаларды пайдаланатын екі желі болса, шлюз

бір желіден екіншісіне хабарламаны бағыттау үшін ғана емес, сондай-ақ түрлі хаттамалардағы хабарламалардың өзінен айырмашылықты өңдеуге қабілетті (біз желі хаттамаларын логикалық деңгейде талқылаймыз). Шлюздер желі шекарасында пайда болады, себебі олар желілердің түрлі типтерін байланыстырады. Бағдарлауыштардан, коммутаторлардан және шоғырлауыштардан айырмашылығы шлюздер желі ішіндегі негізгі құрамдауыштар болып табылмайды. Осы себептен де, шлюздер брандмауэр ретінде де қызмет етеді.

12.5 суретте осы төрт қосатын құрылғылар қалай ерекшеленетіні көрсетілген. Шоғырлауыш жай ғана барлық құрылғылар үшін кез келген хабарламаны береді, осылайша бұл жалпы байланыс (хабарлама тағайындалды ма, жоқ па одан тәуелсіз). Бұл 12.5 суреттің жоғарғы сол жақ бөлігінде көрсетілген, мұнда бірінші (сол жақ шеткі) компьютер шоғырлауышқа хабарлама жібереді, ол содан кейін оны басқа компьютерлерге таратады. Екінші жағынан, 12.5 суреттің сол жақ төменгі бөлігінде көрсетілген коммутатор хабарламаны тек бір ғана тағайындау құрылғысына жібереді, сондықтан байланыс бөлінген. Бағдарлауыш 12.5 суреттің оң жақ бөлігінде көрсетілгендей, шоғырлауыш немесе коммутатор басқасына қосыла алатындай түрде бірнеше жергілікті желілерді байланыстырады. Сызбада жоғарғы желі оң жағындағы екінші компьютер хабарламаны төменгі желідегі сол жақ шеткі компьютерге жібереді. Ережеге сай бағдарлауыш хабарламаны бағыттау үшін желіні анықтауға хабарламаны алушы мекенжайын пайдаланады. 12.5 суреттің оң жағында көрсетілгендей, осылайша көрінетін шлюз бір желіден келесісіне хабарламаның өтуі кезіндегі бір хаттамадан екіншісінен конверсиялай алатынынан басқа, бағдарлауыш сияқты мақсатта қызмет етеді. Содан кейін шлюз бағдарлауыш бір және сол типтегі (хаттама) желілерді бірге жалғаған уақытта алуан түрлі типтегі желілерді қосу үшін пайдаланылады.

Компьютерлік желілер бойынша байланысумен қатар қолданушылар телефон жүйесі көмегімен байланыса алады. Шынында да, бұл телебайланыс үшін ең кең таралған құрал болды.



12.5 сурет Шоғырлауыш (сол жақтағы жоғарғы), коммутатор (сол жақтағы төменгі) және бағдарлауыштар (оң жақта).

Телефон желісі сымдардың тарамдалған жұбы бойынша аналогтік түрде сигналдарды беру үшін бапталған. Компьютер деректері бинарлық түрде, не сымдар ішіндегі электр тоғы немесе дискте не таспада сақталатын магнит заряды түрінде сақталады. Екілік деректер алдымен аналогтік түрге түрленуі керек. Бұл модуляцияны (және сандық сигналды ұқсастыққа түрлендіруді) қажет етеді. 12.6 суретке қараңыздар, мұнда 1001 жүйелілігі (немесе жоғары, төмен, төмен, жоғары тоқ) немесе 1S беру және түрлі үн сияқты адам құлағымен естілетін 0s үшін бір-бірінен қашық болатындай, бір-біріне жақын толқынды түрдегі дыбыстарды аналогтік түрге келтіру қажет. Осындай түрде деректер телефон желілері бойынша беріле алады. Қабылдау құрылғысы аналогтік сигналды қайтадан сандық түрге түрлендіру керек. Бұл демодуляция (аналогтік-сандық түрлендіру). Модем модуляция мен демодуляцияны орындайтын құрылғыны көрсетеді. Қолданушы модемді телефон желісіне жалғайды (мысалы, телефон розеткасы), содан кейін модемі бар тағайындау пунктіне телефон қоңырауын орналастырады. Бұдан әрі, телефон тұтқасы 12.7 суретте көрсетілгендей, MODEM қойғышқа орнатылады. Екі жақтағы компьютерлер енді бір-бірімен өзара әрекеттесе алады. Сөйлескеннен соң телефон қоңырауын аяқтау үшін тұтқа қайтадан кері қойы-

лады. Қазіргі таңда модемдер 1 0 0 1 компьютерлеріне кіріктірілген.



12.6 сурет Сигнал модуляциясы

Бинарлық

Сандық сигнал

Аналогтік сигнал

12.7 сурет Бесік МОДЕМІ

Сондықтан да, сіз өз компьютеріңізді телефон розеткасына қоса аласыз, дегенмен бәрінен бұрын сымсыз карта үйдегі кез келген нүкте деңгейіне тарата алатын сымсыз модеммен компьютердің өзара әрекеттесуіне мүмкіндік береді.

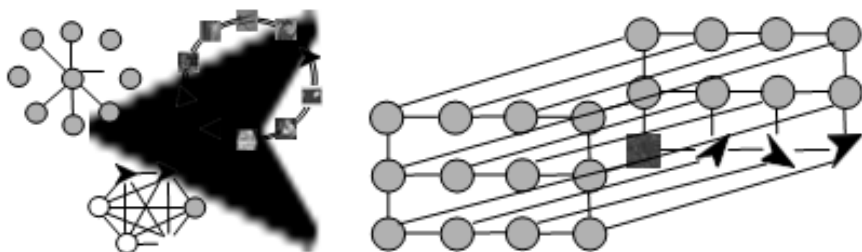


12.7 сурет Бесік модемі

Сондықтан да, сіз өз компьютеріңізді телефон розеткасына қоса аласыз, дегенмен бәрінен бұрын сымсыз карта үйдегі кез келген нүкте деңгейіне тарата алатын сымсыз модеммен компьютердің өзара әрекеттесуіне мүмкіндік береді.

Логикалық деңгейдегі желі

Компьютерлік желілердің бір-бірімен компьютерлік желілер арқылы жалғана алатындай бірнеше тәсілдері бар. Түрлі макеттер бірлесіп желі топологиясы деп аталады. Таңдалған топология желі құнын, сонымен бірге тағайындау орнына жіберілетін хабарлама үшін қажет болатын уақыт мөлшерін шарттайды. Бұдан басқа, топология желі беріктігіне ықпал етеді. 12.8 сурет көптеген түрлі топологияларды көрсетеді. Суретте компьютерлер мен компьютерлік ресурстар (файлдық серверлер, принтерлер, шоғырлауыштар, коммутаторлар, бағдарлауыштар, CD-ROM антенналар, таспалы жинақтауыштар және т.с.с) барлығы дөңгелек түрінде көрсетілген. Тораптарды жалғайтын сызықтар желідегі байланыстарды көрсетеді, ережеге сай шоғырсымнан тұрады (12.1 суретте көрсетілгендей, бұл бұралған қос сымдар болуы мүмкін: коаксиальді шоғырсым, талшықты-оптикалық шоғырсым немесе осы нұсқалардың кез келген әдістері).



12.8 сурет Түрлі желілік топологиялар, тораптар – бұл байланыс тіректерінің компьютерлік ресурстары.

Шиналы топологияның (сол жақтағы жоғарғы), жұлдыз топологиясы (ортаңғы сол), сакиналы топология (жұлдыздың оң жағында), толық өлшемдік (сол жақтағы төменгі) және пішіннің оң жағындағы жақын көршілес топологиялар арасындағы түрлі топологиялары [бір өлшемді (1-D), 2-D, ағаш, 3-D]. Суретте көрсетілмеген – бұл гиперкуб ретінде белгілі 4-D топология.

Желідегі ең қарапайым үлгі нүкте-нүкте желісін береді, мұнда екі ресурс арасында арнайы байланыс бар. Желі топологиясында нүкте-нүкте жиі қолданылмайды, себебі бұл құрылғының бір-бірімен өзара әрекеттестігін шектейді. Біз нүкте-нүкте қосылымы CPU ALU регистрлерін жалғайтын шиналар немесе монитор мен жүйелік тақта арасындағы қо-

сылыс сияқты компьютер ішіндегі жүрістерде екенін көріп тұрмыз. Біз сонымен бірге, желілер арасындағы тік кесінділерден екі нүкте арасындағы белгіленген телефон желілері сияқты қалааралық түрдегі нүктені көре аламыз (мысалы, Ақ үй мен Кремльді байланыстыратын атақты қызыл телефон).

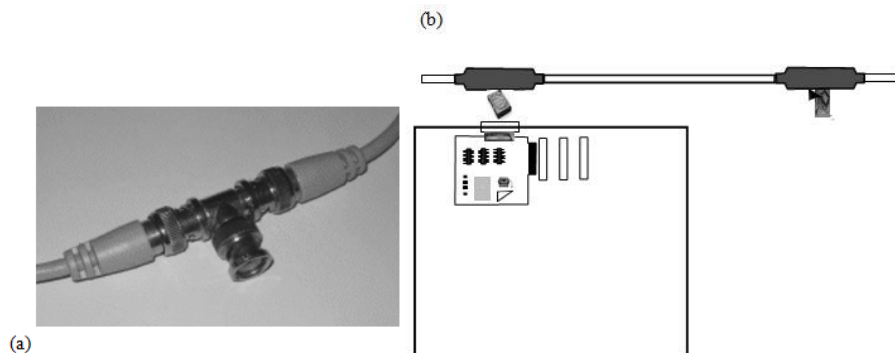
Желі шинасының топологиясы – бұл өте кең таралған көрініс болып табылатын келесі қарапайым үлгі. Бұл топологияда әрбір ресурс бір шоғырсымға жалғанған. Желі шоғырсымы бір ретте бір хабарламаны жеткізеді. Құрылғы шинаға «Т» байланысы арқылы қосылады. Әрбір құрылғы «Т» жалғағышына қосылатын болады, екі шоғырсымға қосылу үшін әрбіреуі келесі құрылғы мен «Т» жалғағышқа апарады. Желінің әр ұшы терминаторға қосылатын болады. Шоғырсымда «Т» байланысының шектелген мөлшері болады, сондықтан желі өлшемі шектелген. 12.9a суретте «Т» жалғағышы көрсетілген және де 12.9b суретте компьютерлердің желілік карта арқылы «Т» жалғағышына қалай қосылатыны көрсетілген. Желілік карта компьютердің жүйелік тақтасындағы кеңею слотына қойылады. «Т» жалғағышы жүйелік блоктың артқы бөлігінен көрініп тұратындай портқа кертілген.

Құрылғының концентратор немесе коммутатор арқылы бір желіге қосылудың басқа да құралы бар. Мұндай жағдайда құрылғы концентратор немесе коммутатордың артқы жағындағы бір жалғағыш арқылы қосылады. Егер желіні кеңейту керек болса, онда концентраторлар мен коммутаторларды тізбекке дәйекті түрде қосуға болады.

Желі трассасында кез келген коммуникацияны тыңдау үшін арналған барлық құрылғыларды бірыңғай шоғырсым арқылы оларға арналмаған хабарламаларды қабылдамауға болады. Шина динамикалық желіні қамтамасыз етеді, оған «Т» жалғағышы болғанша кез келген уақытта құрылғылар қосылуы немесе желіден алынып тасталуы мүмкін. Желі трассасы – барлық желілік топология ішіндегі ең арзаны. Бұл егер ресурс желіден шығарылып тасталатын болса немесе өздігімен пайда болатын болса, сынбайтын желінің сенімді формасы. Біз желінің басқа формалары егер ресурс берілісі төмендесе, желі берілісінің төмендеуіне алып келетінін көреміз. Осыған қарамастан, шинаның топологиясы бойынша желінің көп кемшілігі бар. Барлық хабарламалар бір кабель бойынша өтеді және сондықтан оған қосылған құрылғының көп санына байланысты желі жұмысының тиімділігі нашарлайды. Оған қосылған құрылғы саны көп болған сайын, желі үшін шығындар да көп болады. Бұл өз кезегінде, хабарламалардың үлкен трафигін құрайды. Осылайша, желіде қолдану ықтималы болатын екі құрылғы құрылғының сол бір

уақытта қосылу шамасы бойынша желіге қосылады.

Егер екі құрылғы сол бір уақытта хабарласуға тырысатын болса не болады? Бұл мақұлдау хабарламасы ретінде танымал, ал нәтижесі – хабарламалар бір біріне кедергі жасайды. Осылайша, қақтығыс байқалғаннан кейін, кем дегенде құрылғылардың біреуі тосуы керек, ал ол уақытта басқа құрылғы байланысып жатады. Хабарламаны қақтығыспен өңдеу стратегиясы Интернет технологиясы үшін құрастырылған келесі тарауында талқыланады.



Сурет 12.9 «Т» (а) жалғағышы және желілік карталардың «Т» жалғағышы

Жұлдызды желісі белгілі бір дәрежеде жол желісінің қарама-қайшылығы болып келеді. Жұлдыз желілерінде барлық құрылғылардың бір жалғағыш нүктесі – орталық сервері бар нүкте болады. Бұл сервер байланыс концентраторы, хабарламаны бір машинадан тағайындау орнына дейін бағдарлау ретінде әрекет ететін құрылғыны білдіреді. Жұлдыз тәріздес желінің артықшылығы – бұл оның қарапайымдылығы, құрылғыларды қосу (немесе алып тастау) оңайлығы, сонымен қатар хабарламалардың орнын ауыстыру тиімділігі. Жұлдызды желі жол желілері сияқты хабарлама трафигімен күреспейді (алайда, егер бір уақытта көп хабарламалар болатын болса, концентратор тез арада тар өткізу қабілетіне айналады). Егер хабарлама тек құрылғы концентраторына арналған болса, онда өзінің тағайындалуына қол жеткізу үшін кез келген хабарлама үшін қабылдайтын сілтемелер саны әрдайым біреу ғана болады.

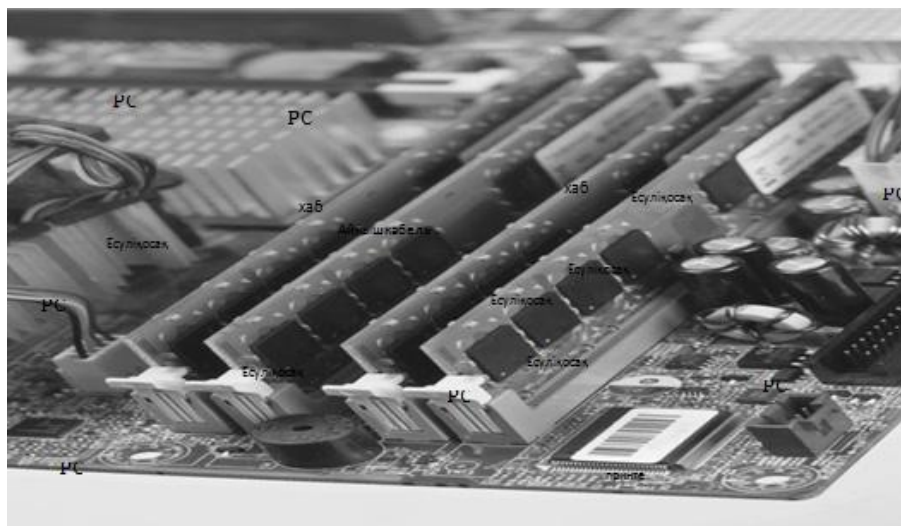
Жұлдызды желінің екі мәселесі бар. Біріншіден, бұл арнайы құрылғыны талап етеді, бұл оны жол желісіне қарағанда, қымбатырақ желіге айналдырады. Жұлдызды желі концентраторы желіде суреттелгендей

аппараттық деңгейдегі концентратор немесе коммутатор болуы мүмкін. Жұлдыз тәріздес желінің күпшегі сонымен қатар желіні жұлдызды және аса қымбатырақ жасайтын сервер болуы мүмкін. Екіншіден, бір құрылғының жоғалуы желіні нашарлатпайды, егер концентратор жоғалатын болса, барлық құрылғылар қосылуларын жоғалтады.

Бүгінгі күні жергілікті желілерді құрудың жалпы әдістемесі – бұл бірге жұмыс істейтін жұлдызды желіні қосу. Бұл тораптарды бірге дәйекті түрде қосу жолымен орындалады. Мысалы, концентратор бір желідегі ресурстарды бірге қосу үшін қолданылады және концентратор өз бетімен екінші желінің ресурстарын бірге қосатын басқа концентратормен қосылған. Бұл барлық құрылғылар бір бірімен өзара әрекеттесе алатын үлкен бірыңғай желіні құрайды. Мұндай жағдайда егер ресурстар әртүрлі күпшеге жалғанған болса, олардың кейбіреулерінің арасына талап етілетін қосымша беріліс болады. Концентраторлардың (сонымен қатар коммутаторлардың, бағдарлауыштардың және шлюздердің) қосылудың шектелген саны болатын болса, онда мұндай әдістеме желіге оңай өсуге мүмкіндік береді. Керегі бұл қосымша концентраторлар. 12.10 суреті осы концепцияны суреттейді. «Айқыш шоғырсым» екі концентраторларды бір қосу үшін қолданылады. Берілген нақты сызбада желідегі құрылғылардың барлығы концентратормен сымдардың байлау жұбы бойынша байланысқан.

Сақиналы желіде құрылғылардың өздерінің көршілері бар қосылу нүкте түрі болады. Құрылғылар арасындағы байланыс хабарламаны көршіге жіберуді талап етеді және көрші хабарламаны өзі тағайындау пунктіне жетпегенше сақина бойынша жібере береді. Сақина кез келген екі торап арасында жаңа құрылғыларды қосумен оңай кеңейе береді. Алайда сақиналар көп болған сайын, хабарлама өтетін ықтимал қашықтық та өсе береді. Бұл өз кезегінде берілістің созылмалы уақытын құруы мүмкін. Сақиналар жол желілеріне қарағанда арзанырақ, себебі орталық нүкте ретінде қызмет етуге бөлінген ресурсі жоқ. Сақиналы желі хабарлама жүретін бағыт тарапынан әмбебап бағытталған немесе екі бағытты болуы мүмкін. Егер құрылғы автономды режимге көшетін болса, бір бағытты сақина желіні кейбір құрылғылар басқалармен байланыса алмайтындай екі бөлікке бөліп тастайды. Мысалы, топология бұл 12.8 суретте көрсетілген сақина, бір бағытты болды және барлық коммуникациялары сақина бойында сағат тілімен жүрді деп болжамдайық.

Принтер



Сурет 12.10 Хабтармен жалғанған екі жол желілері

Егер жоғарғы құрылғы қатардан шығатын болса, онда тоқтап қалған құрылғыдан тікелей сол жағында тұрған құрылғы кез келген басқа құрылғы дейін жете алмайды, ал тоқтап қалған құрылғыдан тікелей оң жағында тұрған құрылғы кез келген басқа құрылғыдан хабарлама қабылдай алмайды. Екі бағытта сақина желі бөліктері оқшауланғанға дейін екі құрылғының жоғалуын талап етеді.

Ұялы желі де әрбір құрылғының қосылу нүктесі – басқа әрбір құрылғысы бар нүктесінің болуын талап ететін толығымен жалғанған желі ретінде танымал. Егер желіде N құрылғылар болатын болса, онда 2 қосылулар сияқты $(p - 1)$ көп болады. Кез келген құрылғы хабарламаның трафигі туралы ойланбай тікелей кез келген құрылғымен өзара әрекеттесе алады және егер құрылғы алынып тасталса, ол желінің байланысы зиян келтірмейді. Алайда ұялы желі бағасына байланыс шектеулі болып келеді. Онымен қоса, бұл аса көп қосуларды талап етеді, желі әрбір құрылғының осы қосуларды жалғау үшін $p-1$ қолжетімді порттарының болуын талап етеді. Ұялы желі негізінен жоғары өнімділігі қосарлама өңделген компьютерлерде болады, бірақ компьютерлік желілерде қолданылмайды. Алты құрылғыдан тұратын кішкене ұялы желі 12.8 суретінде көрсетілген. Алты құрылғының жалпы 25 қосулары бар. 50 құрылғы желісі

үшін қадет қосулар санын елестетіп көріңіз. Ол үшін 2401 қосулар қажет болады.

Сакиналы желі желінің жақын көршісі формаларының бірі болып табылады: көршілер бір - бірімен тікелей байланысқан құрылғыларды білдіреді. Ұялы желі әрбір құрылғы кез келген құрылғыға қосылуына байланысты жақын көршінің түпкілікті формасы болып табылады, себебі әрбір құрылғы жақын көрші болып саналады. Желінің жақын көршілерінің көптеген аралық формалары болады. Оларға желі ағашы жатады: бір құрылғы ағаштың түбірлі торабы болып табылады және ол бір немесе бірнеше еншілес тораптарға жалғанады, және әрбір бала бір немесе бірнеше тораптарға жалғанады, және тағы сол сияқты. Жалпы ағаш әрбір тораптың екі тармағы болатын бинарлы ағашты білдіреді. Егер желінің бинарлы ағашында n деңгей болатын болса, онда торап $2^n - 1$ болады және хабарлама көзі мен тағайындау тораптары арасындағы трансмиссияға қарағанда $2 * (n - 1)$ көбірек талап етеді. 12.8 суретінде желінің жақын көршілерінің төрт формасы көрсетілген, 1-D А желісі, 2-D А желісі, 3-D желісі(толығымен өшпеген) және желі ағашы. Ал 4-D желісі гиперкуб деп аталады.

Топологиядан басқа, біз желіні олардың физикалық өлшемдеріне қарап жіктей аламыз. Өлшем ресурстар арасындағы қашықтық сияқты ресурстар санының шамасы болып табылмайды. Бүгін желінің аса кеңінен таралған өлшемі ЖЖ (жергілікті желі) болып табылады. Жергілікті желі жалғанған (компоненттер есулі қосақ, коаксиальды шоғырсым немесе талшықты – оптикалық шоғырсым сияқты физикалық ортаның көмегімен қосылған) немесе сымсыз желі (компоненттер не инфрақызыл жарықтың, не радиосигналдардың кейбір түрлерінің көмегімен өзара әрекеттеседі). Көптеген жергілікті желілер – бұл желілік байланыс немесе сымы бар не сымсыз байланыс үйлесімі. База сымсыз жергілікті желі таралған құбылыс болып табылмайды.

Бүгінгі күні көптеген жергілікті желілер кішкене жергілікті желілер топтамасын білдіреді. Мысал ретінде университетті қарастырайық. Университеттің компьютерлік зертханалары болады. Қандай да бір зертханалардың бірінде бір ЖЖ бар. Зертхана ресурстары бір бірімен концентратор немесе (аса ықтимал) ауыстырып-қосқыштар сияқты кейбір хабарды кеңінен беретін құрылғылар көмегімен жалғанған. Егер бөлме жеткілікті кең болса, онда бірнеше концентраторлар немесе коммутаторлар бір - бірімен жалғануы мүмкін. Негізі егер өте көп құрылғы болатын болса, онда бірнеше коммутаторға жалғанатын бағдарлауыш қолданылуы мүмкін. Бұл бір бөлме аясында бірнеше жергілікті желіні құруға мүмкіндік береді.

Университеттің ғимарат екенін ескеретін болсақ, онда ол әрине бірнеше компьютерлік зертханалардан тұрады. Әрбір зертхана (бөлме) басқа зертханалармен бағдарлауыш арқылы жалғанған. Бұл бізге жергілікті желіні береді. Бұл жергілікті желілер ғимараттағы басқа жергілікті желілерге қайтадан бағдарлауыш және мүмкін шлюздер арқылы жалғанады. Мысалы, ол жерде ғимараттың бірінші қабатында, машинада, үйде, терезеде төрт жергілікті желі және екінші қабатта екі жергілікті желі болуы мүмкін, бұл Macintosh үй компьютерлері. Бірінші төрт жергілікті желілер бір - бірімен бағдарлауыш көмегімен, сонымен қатар екіншімен өткізілген екі зертхана бір - бірімен бағдарлауыш көмегімен жалғанған. Екі бағдарлауыш бір бірімен шлюз арқылы қосылған. Аудиториялар мен факультет кеңселерін бірге қосу үшін ғимараттың басқа қабаттарының өздерінің жергілікті желілері болады. Осылайша, ғимаратта жергілікті желі болады. Және бұл ғимарат өз кезегінде басқа ғимараттармен жалғанады. Барлық студенттік қалашық жергілікті желілердің жергілікті желісін білдіреді.

Атауына байланысты университеттік қалашық сияқты қандай да бір ұйымның сайтын қамтитын желі болуы мүмкін (кампус ауқымындағы желі). Бұл алдыңғы тарауда суреттелген сияқты жергілікті желілердің жергілікті желісі екені анық. Сонымен қатар, егер кампустың қалада бөлімшесі болса, онда ол аймақтық облыс шегінен де шығуы мүмкін. Жергілікті желінің басқа полюсінде ол ПЖ (Персоналды желі) деп аталады. ПЖ – адамның үй шаруашылығындағы ресурстардың бірігіп қосылған кішкене жергілікті желісі. Үй компьютерлерінің көптеген тұтынушыларының қазір барлық үй компьютерлері мен принтерлерінің бір - бірімен жалғанған желілері бар. Олар толығымен сымсыз қосылу немесе мысалы принтер сияқты кейбір ресурстар арқылы жалғануы мүмкін. Осы құрылғылардың қосылу құрылғылары желілік коммутатор немесе концентратор болуы мүмкін, немесе бұл концентратор ретінде қызмет ете алатын модем болуы мүмкін.

Желі хаттамасы қолданатын жергілікті желі (Желі хаттамалары тарауын қараңыз) интражелі ретінде танымал. Интражелінің артықшылығы Интернетті қолдану үшін арналған қосымшалар (электрондық пошта, веб-браузер, Telnet / SSH және тағы басқалары) сыртқы сигналмен қалай жұмыс істейді, іште де солай оңай жұмыс істейді. Экстранет бұл ішкі желіге қолжетімділік өшіріліп тасталған сайттармен орындалатындай басқа желілермен салыстыру бойынша интражеліні кеңейту. Бұл үшін өшіріліп тасталған пайдаланушыларға іс жүзінде рұқсат етілгеніне кепілдік беру үшін қосымша қауіпсіздік қажет. Негізінен, жүйеге кіру үдерісі қолданылады. Виртуалды меншікті желі (VPN) сыртқы желінің

таралған түрі болып табылады. Виртуалды меншікті желісі ішкі желі шегінен шығатын хабарламалары тіпті егер пакеттерді талдау (бұл осы тарауда одан әрі талқыланады) сияқты әдістердің көмегімен ұстап қалса да, түсініксіз болатындай шифрлеу технологиясын жиі қолданады.

ЖЖ және ПЖ олар ресурстарды жақын арада (футтарда, мүмкін миллион қашықтығында) қосуына байланысты кішкене желі болуы ықтимал. Толығымен қаланы немесе оның бөлігін қамтитын қала желісіндегі (ҚЖ) үлкен желі. Кішкене аралықта адам МС анықтаудан басып озады. Мысалы, бірнеше шаршы миллионға созылған университеттер МС құрамында емес, ҚЖ болады (бұл жерде айырмашылығының мәні жоқ). Үлкен аралықта, адам бүкіл қаланы басады. Мұндай жағдайда талшықты-оптикалық шоғырсымның жоғары жылдамдықты физикалық қосулары немесе радиода пайдасыз болуы мүмкін. ҚЖ Лондон (Ұлыбритания), Женева (Швейцария) және Сан-Франциско (Калифорния, АҚШ) сияқта қалалардан табуға болады. Біз ЖЖ (Жаһандық желі) құру үшін МС шегінен тыс желілердің қамтуын кеңейтуімізді жалғастыруға болады. Олар округ, штаттар, мемлекеттер немесе бүкіл әлем арқылы ресурстарын қоса алады. Осындай ЖЖ бірі «Sohonet» деп аталады, ол әртүрлі өндірістік желілерді және бүкіл әлем бойынша (Лос-Анджелестен Нью-Йоркқа, Канадаға, Жаңа Зеландияға және көптеген еуропалық қалаларға) компаниялардың постпродакцияларын біріктіреді. Ең үлкен жаһандық желі Интернет.

Біз жергілікті желі үшін жүзеге асыру формалардың біреуін қарастырайық. Аса танымал жүзеге асырушылардың бірі Ethernet желісі деп аталады. Біріншіден, 1970 жылдардың басында өңделген Ethernet коммерциялық релизінің түпнұсқасы 1980 жылы 3Com компаниясымен шығарылды, Ethernet 1982 жылы стандарт берілді және Token Bus және Token Ring желі формаларына бәсеке болды. Алайда, басында коаксиальды шоғырсым бойынша шина топологиясын қолданумен (Т жалғағышымен) іске асырылды, қазіргі желілері есулі қосақты және концентраторларға немесе коммутаторларға қосылған құрылғылары бар талшықты-оптикалық шоғырсымды қолданылуы мүмкін. Ресурстарды тікелей концентраторға немесе коммутаторға қоса отырып, бұл тек тиімді ғана емес, сонымен қатар орнатуға кететін шығынды азайтады және желіні басқаруды жақсартады.

Ethernet технологиясы көптеген жылдар бойы 10 Мбит теориялық өткізу қабілетінен 100 Гбит ағымдағы жоғары өткізу сызығына дейін жақсарды. Ethernet бірқатар желілер концепциясын енгізді, олар негізінен қазір де кездеседі. Олар қақтығысуды бақылау механизмдерінен, Ethernetпетрансляторларынан, мекенжайлар көздері мен тағайындалуы

үшін арналған 48 битті MAC дербестендіру сызбасынан (беріліс ортасына қол жетімділікті басқару), Ethernet кадр пішімінен, механизмдер қателіктерін өңдеуден, сонымен қатар көптеген әртүрлі компьютерлер түрлері Ethernet қосыла алатындай әртүрлі адаптерлерден тұрады. Бұл жерде біз қақтығыстарды бақылау сызбасын ғана қарастырамыз.

Ethernet енгізілген қақтығыстарды бақылау формасы қайшылықты бақылауы бар (CSMA / CD) Carrier Sense Multiple Access деп аталады. Біріншіден, құрылғы хабарламаны трансмит үшін дайындайды. Ол желілік медианың бос еместігін сезуге тырысады. Егер бос болмаса, бұл құрылғылар тасымалдаушы босағанын күтпейді. Содан кейін ол өз хабарламасын тасымалдаушыға орналастырады. Бірақ, егер басқа құрылғы тосып отырса, онда екі құрылғы да сол немесе сол сәттегі медиаға хабарламаны орналастыра алады. Сондықтан, тіпті егер ол жіберетін болса, онда егер кез келген басқа хабарлама бұқаралық ақпараттар құралдары арқылы келетін болса, құрылғы да жіберуге тырысады. Егер жіберетін болса, онда қақтығыс байқалады. Құрылғы тез арада мәліметтерді жіберуді тоқтатады және кептеліс тұрады сигнал жібереді. Джем сигналы сол сәтте жібермеу үшін басқа ресурстарды ескерту үшін қолданылады. Варенье сигналы да солай, себебі ол да хабарламаны басатын іс жүзіндегі хабарламадан тұратын тасымалдаушы арқылы жіберіледі. Кез келген қабылдау құрылғысы кептелу туралы сигналды қабылдайды және сіз ол хабарламаны қабылдағанын білесіз, бірақ ол зақымдалған болады. Бүгінгі күні, екі жіберетін құрылғы да оларды қайтадан беру әрекетіне дейін біршама уақыт күтеді. Айырып-қосқыштар Ethernet концентраторлар және автобус аялдамалары арқылы қосылған кезде, қақтығыстар азайды, бірақ CSMA / CD қолданыла берді.

Біз сонымен қатар желіні желідегі компьютердің рөлі сияқты рөл бойынша жіктей аламыз. Жиірек, біз желіні тең – теңге немесе клиент – серверлік желі түрінде аламыз. Бір дәрежелік торап әрбір компьютер шамамен әрбір басқа компьютерге тең екенін білдіреді. Бұл желіні клиент – сервер үлгісінен ажыратады. Клиент ақпаратты басқа компьютерден сұрайтын компьютерді білдіреді. Сервер сұранысты қабылдайтын және сұратылып отырған ақпаратқа жауап беретін компьютерді (немесе құрылғыны) білдіреді. Содан кейін желілік клиент – сервер бір немесе бірнеше серверден тұратын желіні білдіреді. Тең желілер арзан болады, алайда күшті сервер қажеттілігі көптеген желілердің клиент – сервер үлгісіне ереді. Серверлер көптеген жағдайларда аса қымбат компьютерлер болып табылады, себебі олар басқа компьютерлік желілерге қарағанда қатқыл дискіде көп сақтауды және жауап қайтарудың тез уақытын талап етеді. Серверлер жиірек бірліктің, мүмкін бірнеше қатқыл дискілері

бар, үлкен жүйесі ретінде көрінеді. Үлкен файлдық серверлер бұл 12.11 суретте көрсетілгендей шкафта бекітілген кішкене блоктардың топтамасы.

Қалаулы сервис түріне байланысты серверлердің бірнеше түрлері болады. Сервердің бір түрі бұл файлдық сервер. Мұндай жағдайда сервер рөлі клиент талаптары бойынша желі арқылы файлдарды жіберу болып табылады. Типтік файлдық сервер сұраныстарға жергілікті желі арқылы және қызметтерге тек компьютердегі жергілікті желі арқылы жауап береді.



Сурет 12.11 Файлдық сервер бағанына монтаждау үшін

Файл – серверлер қолданбалы бағдарламалық жасақтамаларды және мәліметтер файлдарын қолдау үшін қолданылуы мүмкін. Яғни, клиенттер тек өздерінің меншікті операцияларын сақтайтын болады. Бағдарламалық жасақтаманы іске қосуға ынталы тұтынушыға талап бойынша желі арқылы бағдарламалық жасақтаманы жүктеу керек болады. Немесе файлдық сервер аз ғана біріккен қолданбалы бағдарламалық жасақтамамен және / немесе мәліметтер файлдарымен шектелуі мүмкін.

* Сервердің басқа формасы веб-сервер болып табылады. Бұл мәліметтер файлдарын (веб-парақтарды) және бағдарламаларды (скриптерді)

ұзақ сақтайтын файлдық сервердің ерекше түрі. Алайда, веб-сервер тек жергілікті желідегі ғана емес, сонымен қатар Интернеттегі кез келген орыннан клиенттердің сұраныстарына, сондай-ақ арнайы HTTP (гипермәтіндерді беру протоклы) сұраныстарына жауап бере алады. Жауаптары HTML-файлдары, файлдық серверде сақталатын құжаттар немесе CGI (Common Interface шлюзі) скриптері арқылы сервермен динамикалық түрленетін веб-парақтар түрінде болуы мүмкін. Веб-сервердің бірнеше міндеттері бар: файл-сервер, онымен қоса сценарийлер, сұраныстар мен қателіктерді тіркеу сервері жағындағы файл-сервер жұмыс істемейді, сонымен қатар қауіпсіздікті өңдеу. Сервердің тағы бір формасы мәліметтер базасының сервері болып табылады. Ол клиенттердің мәліметтер базасына сұраныстарына мәліметтер базасынан алынған жауаптармен бірге жауап қайтарады. Онымен қоса, сервер мәліметтер базасын басқару жүйесінен алынған мәліметтер негізінде есептерді түрлендіруі мүмкін. Файлдық сервер сияқты, мәліметтер базасының сервері тек жергілікті клиенттерге жауап қайтарады. Файлдық сервер сияқты мәліметтер базасының сервері егер сервер аса кең желімен қолжетімді болса, алыс жерлердегі тіркелген клиенттерге жауап қайтаруы мүмкін. Бірақ, веб-сервер сияқты мәліметтер базасы серверінің міндеттері файлдарды жіберу аясынан тыс шығады. Басқа серверлер басып шығару серверінен, пошталық серверден және FTP серверлерден тұрады. Басқа серверлерге қарағанда, басып шығару сервері файлды қайтармайды, бірақ оның орнынан басып шығару тапсырмасын бақылайды және басып шығарылу аяқталды тапсырмасын анықтаумен немесе қате туралы хабарламамен жауап береді. Пошталық сервер жағдайында біреуі электрондық поштаның басқа серверіне жіберу үшін электрондық поштаны серверге жібереді. Электронды поштаның екі сервері бір бірімен қатынасады, және содан кейін электрондық пошта тағайындалу орнынан жеткен соң, қабылдаушы сервері тұтынушыға қабылдаушының жаңа электрондық поштасы бар екенін хабар береді. FTP сервері файлдар үшін сұраныстары мен файлдардың жауаптарымен веб-сервер немесе файлдық серверге кішкене ұқсас.

Желілік протоколдар

Протоколбасқару үшін бекітілген, адамдар өздерін қалай ұстайтыны және бір бірімен қалай өзара әрекеттесетіні туралы ережелер жиынтығы болып табылады. Бұл дипломатия немесе этикет формасы, немесе құрал ретінде қарастырылуы мүмкін, оның көмегімен зерттеуші басқа зерттеушілердің нәтижелерін хабарлай алады. Компьютерлік желі бір-

неше әртүрлі қабаттарда болады. Біз байланысты қабылдайтын медиа және физикалық формаларды талқылағанда, желі аппаратық деңгейде жұмыс істейтіні туралы ойлануымыз мүмкін. Біз желіні ақпараттар қателіктерін өңдеу арқылы сенімділікті қамтамасыз ете алатынын талқылай аламыз. Немесе біз қолданбалы бағдарламалық жасақтама трансляция үшін хабарлама дайындайтыны туралы болжамдай аламыз. Ал негізінде, осы қадамдардың әрбіреуі қажет, және біз тек олардың қалай жұмыс істейтінін ғана емес, сонымен қатар өзара әрекеттесу кезеңдерін де анықтауымыз керек. Яғни, біз желідегі байланысты қабаттар қатары ретінде қарастырамыз және хабарлама қабаттан қабатқа ауысатындай бір формадан басқа формаға ауысуы керек. Желілік протокол желі қабаттары бір бірімен өзара әрекеттесетін ережелерді қамтамасыз етеді. Бұл өз кезегінде әртүрлі желілер бір бірімен мәліметтермен алмаса алатын ережелерді қамтамасыз етеді. Бұл сонымен қатар желілік программистерді және әкімшілікті желіні басқаруды жүргізу және қызмет көрсету туралы хабарлайды.

Физикалық деңгейде аса таралған протокол Ethernet протоколы болып табылады, ол қолданылатын кабельдер түрлері, қатығысты өңдеу механизмі (CSMA / CD), сонымен қатар топологияның қолжетімді түрлері: шиналар, жұлдыздар, ағаштар сияқты желің аспектілерін анықтайды. Басқа қолданылатын танымал протоколдар тікелей аудио және видео берілісті ұстайтын LocalTalk, Apple, Token Ring өңделген компьютерлері (IBM компаниясымен өңделген) және ATM (асинхронды беріліс режимі).

Аса жиірек қолданылатын желілік протоколы TCP / IP (берілісті басқару протоколы / желіаралық протокол), ол мәліметтермен Интернет арқылы алмасады, барлық компьютерлердің талаптары болып табылады. Осылайша, компьютерлер басқа желілік протоколдарды қолдана алатынына қарамастан, олар сонымен қатар TCP / IP жұмыс істеуі керек. Ашық жүйелердің (OSI) өзара әрекеттесу үлгісіне балама ретінде барлық желілік коммуникациялар үшін арналған стандарт ұсынылды. Осылайша, TCP / IP нақты протокол болуына қарамастан, OSI жаңа желілерді құрастырушылар үшін нысана ретінде қолданылады. Бүгін TCP / IP сияқты, OSI да негізінен желілік протоколдар үшін үлгілері келтіріледі, сондықтан біз оларды осы жерде қарастырамыз.

OSI үлгісі 12.1 кестесінде көрсетілгендей жеті қабаттан тұрады. Әрбір қабаттың 7 санымен нөмірленетін ең жоғарғы қабатынан бастап, 1 санымен нөмірленетін төменгі қабатына дейін өз нөмірі болады. Жоғарғы төрт қабат қабылдаушы қабаттар ретінде танымал, себебі бұл деңгейлердің қызметі басты компьютерде орын алады. Бұл қаптау қабаттары

бірігіп беру үшін хабарламаны немесе хабарламаны қабылдауды және оны талдауды түзеді. Төменгі үш қабаты медиа қабаттары ретін таны-мал, себебі бұл қабаттарда мекенжайға жіберілетін, қолданылатын және физикалық жеткізілген хабарламалар болады. OSI нақты протокол болмағанына қарамастан, ол желілік протоколдарды, онымен қоса, Жалпы басқару ақпаратының протоколын, X.400 электрондық поштасын, X.500 қысу каталогын айырбастау және IS-IS (Шлюз аралық жүйе) бағдарлауыш протоколын құру үшін қолданылды.

1 қабат, физикалық деңгей болып табылатын ең төменгі қабат. Бұл қабат байланысқысы келетін (мысалы, компьютер немесе концентратор) және беру ортасы бойынша байланыс орнататын (коаксиальды кабель, талшықты – оптикалық кабель және сол сияқты) құрылғы ретінде анықталады. Бұл қабат беріліс үшін және сигналды модуляциялау үшін (қажеті болса) қажет кернеу сияқты заттарды, TCP / IP талап етеді, негізінен бұл протоколдар жинағы кейде протоколдарды айдау ретінде аталады.

12.1 Кестесі OSI Моделі Топтар

Мәліметтер блогы деңгей номері мен функциясының атауы

7. Мәліметтерді қолдану бұл қолданбалы бағдарламалық қамтамасыздандырумен бірге пайдаланушы арасындағы арақатынасы

Көрініс 6. Мәліметтерді көрсету,шифрлеу/дешифрлеу

5. Байланыс деңгейінің түйіндер сеанстары, сессиялармен басқару

Сегменттер 4. Транспорттау. Ағынмен басқару және сенімділік

Packet / Datagram 3.Желілі логикалық дербестендіру,маршруттау

Жиектеме 2. Мәліметтерді дербестендіру кезінде беріп тұру каналы

Бит 1. Физикалық иеленуші, екі кодта дабылды жіберу мәліметтерді жіберу ортасында қалай орнату және қосылымды қалай тоқтату туралы.

Байланысты қосу кезінде, қабат трафик хабарламасын анықтау мүмкіндігін қажет етеді (бәсекеге тұратын) және трафиктің болуы кезінде шешу құралын қажетсінеді.Мәліметтерді жіберу кезінде, бұл қабат мәліметтерді 2 қабаттан алады және оны формаға қайта қалпына келтіруге міндетті, ол беріп жіберу үшін үйлесуі керек.Мәліметтерді алған кезде, бұл қабат мәліметтерді беріп жіберу ортасынан алады (электр тогы,дыбыс толқыны,жарық импульстері) және 2 қабатымен бірге бірлесіп пайдалануы үшін екілік форматта мәліметтерді қайта құрады. Бұл жағдайда бұл қабатта хабарлама бит жүйелілігі түрінде беріледі. 1 қабатты жүзеге асырудың бірнеше түрі бар,оған IEEE 802.3, IEEE 802.11, Bluetooth, USB және концентраторлар қосылады.

802 дегеніміз не?

Мүмкін Сіз жазба түрде шарттарды көрген шығарсыз,олар IEEE 802.

xx сияқтылары ол нені білдіреді? IEEE Электртехника және электроника инженерлерінің Институты болып келеді, олар бірнеше қатар стандартты әртүрлі тырысулар арасында ұйымдастырылды, яғни олар оны құрды. Алайда заңсыз және талаптарсыз стандарттар пайдаланушылардың көбі тұрақталған стандарттарға сәйкес болуына тырысуын білдіреді, яғни олардың тырысушылықтары пайдалануға жарамды және қолданысқа ие болатынына кепілдеме береді.

Телекоммуникациялардың көп стандарттары алға шығарылады және олар IEEE802.xx, IEEE ретінді белгіленеді мұндағы 802 «1980 жылдың ақпан айы» бірінші айы үшін дегенді білдіреді, IEEE телекоммуникациялық стандарттарды талқылау үшін кездестірді дегенді білдіреді. (бұл сонымен қатар еркін бірінші номер болды). Стандарттар ішінен кейбіреулері не, IEEE 802 астынан қайсысылары жүреді?

- IEEE 802.3-Ethernet
- IEEE 802,7-Broadband LaN
- безопасность IEEE 802.10-Lan
- IEEE 802.11 сымсыз локальды жүйе (түрлердің саны шығарылған болатын, әрбір берілген хат, 802.11 б, г, 802,11 и 802,11 ретінде)
- IEEE 802.15.1 стандарты бойынша сертификаттау-Bluetooth
- 802.22 сымсыз жүйе IEEE аймағы
- IEEE 803,23- жұмыс топтары үшін апатты- қорғау қызметтері

IEEE 802 IEEE стандарттарынан тыс бұл жағында басқа да стандарттар топтарын құрды, компьютерде қалқу нүктесінен көрсетіліп басталатын (IEEE 754)POSIX (Portable Работа жүйе интерфейсінің жұмысы ING) стандартына қарай, Unix және Linux жүйелері бағдарламалау инженериясы (IEEE 610) облысында стандарттаға (IEEE 1003) жауап береді.

2қабат деңгейі деректерді беру арнасыболып табылады. Егер хабар сол желідегі құрылғыға арналған болса (мысалы, сол, бір концентратор немесе коммутатордың арасын бөлетін), қарым-қатынас осы деңгейде жүргізіледі, және 1 қабатынталап етпейді. Бұл деңгейде деректер блок ретінде қалыптастырылады, ол кадрлар деп аталады. Жақтау кадрлардың синхрондауын көрсетеді, хабарлама басында биттердің жүйелілігін білдіреді. Бұл қабат құрамында екі жақасты қабаты болады. Жоғарғы басқару жақасты қабатының логикалық арнасы (LLC), деңгейасты. Бұл қабат мультиплекстеуді қамтамасыз етеді, ол бірнеше қабаттасатын хабарларды бір мезгілде өз мойнына алу мүмкіндігін білдіреді.

Мультиплекстеу арқылы, мүмкін, бірнеше пошталық хабарламалары бар әр түрлі желілік хаттамаларды иеленеді, олар сол уақытта бірге болады, бұқаралық ақпарат құралдарының сол желісін бөлісіп отырады. Төменгі деңгей MAC. MAC-мекен-жайы осы желі шегіндеқұрылғылар-

дың орналасқан жерін белгілеу үшін пайдаланылады . MAC деңгейі құрылғыларына бір желі ішінде бір-бірімен араласуға мүмкіндік береді, тек олардың MAC мекен-жайларын пайдалана отырып. Бұл жақастық қабаты LLC жақасты қабаты және физикалық деңгей арасында интерфейсі ретінде қызмет етеді. 2 Қабаты өткізуі өзіне IEEE 802.2, IEEE 802.3, PPP (нүкте-Point Protocol), X-25 алмасу коммутаторының пакеттерін қамтиды, сондай-ақ банкоматты. Ethernet 1 және 2 екі топтары үшін іске асыру болып табылады.

3 қабат желілік деңгейін білдіреді. Бұл қабатта, желінің физикалық сипаттамалары мәселе болып табылмайды. Оның орнына операциялармен тағайындалған бұл қабаты деректерді ауыспалы тізбектер ұзындығына байланысты хост және мекенжайы ретінде қарайды. Бұл қабаттың жүріп тұруы операциялардың бағдарлауын өңдейді . Құрылғыға түсіп жатқан хабарлар, оның жұмысы келесіні білдіреді, яғни хабарламаны алға жіберу үшін алушының мекенжайын қарастырады өзінің бағдарлау кестесінде жазулары бар, ол хабарламаны қай желіге орнату үшін анықталады.

Сондықтан маршрутизаторлар осы деңгейде жұмыс істейді. Бұл қабатта, хабарлама компоненттері жекелеген пакеттерге қалыптастырылады. Хабарлама таратушы құрылғымен тасымалданады, ең алдымен, бірнеше пакеттерден тұрады, бәлкім, ондаған немесе жүздеген, көлеміне байланысты хабарлар. 3-ші деңгейдегі жүзеге асырылу өзіне келесілерді қамтиды IP-мекен-жайы (TCP / IP және одан төмен), AppleTalk, IPX (Internet Packet Exchange), ICMP (Internet Control Message хаттамасы) және ARP (Address Resolution Protocol).

4 қабат транспорттау деңгейі болып табылады. Оның ең басты міндеті келесіде қамтылады, бөлек пакеттердегі физикалық тасымалдауы мен хаттаманың жоғарғы деңгейі арасындағы айқындықты қамтамасыз ету. Бұл деңгейде хабарламалар сегмент түрінде берілген, олар әлдеқайда ұсақ бірліктерге (мысалы, пакеттер) бөлінетін болады. Бұл деңгейде көрсетілетін қызметтері арасында ағынмен бақылау және сенімділік болып табылады. Сенімділік үшін бұл қабат пакеттің еш қатесіз алынғандығы туралы кепілдеме беруі керек. Сенімді өңделу үшін бөлшектер төменгі жағында сипатталған. Ағынмен басқару келесіні білдіреді, екі құрылғы бір-бірімен өзара әрекет етеді, БАҚ арқылы әртүрлі жылдамдықпен араласады, мысалы екі компьютерде модемнің әртүрлі жылдамдығы болады.

Сенімділікті өңдеу үшін, 4 қабаты кепілдік беруге тиіс, бұл пакеттер жоғалып кетеді, жолда ауыстырылады. Осын өндеуге арналған түрлі тетіктері бар. Қарапайым тәсіл бұл әрбір пакетке мөр басу жалпы хабар-

ламадаоның реттік нөмірін көрсету 4.7 сияқты. Қабылдау құрылғысы келу үшін жеті пакеттерін күтеді. Егер барлық пакеттер келсе, 4 нөмірінен басқасы, тапсырушы құрылғы оны қайтадан беруге міндетті. Сонымен қатар, әрбір пакетте қателері табылғандығы туралы ақпарат-мазмұндалуы қажет, бақылау сомасы сияқты қателер. Бақылау сомасы екі мәнді негізінде есептеуді білдіреді, хабарлама ішінде тұрады. Тағы да бір қарапайым есептеу үшін бақылау сомасына хабарламадағы 1 бит санын қосу. Мысалы, егер бинарлы хабарлама 256 бит тұрса, және 104 бірі болып табылады 1s, ал қалған 152 0s, онда бақылау сомасы 104. Дегенмен, әдетте, бақылау сомасы өлшемді белгіленгендей болуы тиіс. Осы мақсатқа жету үшін бақылау сомасының функциясы 1 битті қатарда қосу мүмкін, содан кейін ол алдын-ала таңдалған мәнге оны тара-тады. Бақылау сомасы бөлуден болатын қалдық болып келеді (бұл мод, немесе модуль бойынша, оператор). Модты пайдалану ойы хэш-функ-циясы ретінде белгілі. Бақылау сомасының бірқатар түрлі алгоритмдері бар, оған саусақ іздері, кездейсоқ функциялар мен криптографиялық функциялары. Кез келген жағдайда пакетін алғаннан кейін, қабаты 4 келесіні анықтайды, қандай да бір мәліметтер пакетте қате болып келе ма бақылау сомасының мәнін пакеттегі деректермен салыстыру жолымен. Егер қате табылса, қабаты 4 сұрау салу жібереді, сондықтан пакетін қай-та қойылуы мүмкін. 4-ші іске асыру деңгейін қамтиды TCP, UDP (екеуі де бірге талқыланады TCP / IP төмен) және SCTP (Stream Protocol беру-мен басқару).

5 қабаты сеанс деңгейі болып табылады. Бұл қабат екі құрылғылар ара-сындағы байланысын қамтамасыз етеді. Екі құрылғылар деректерімен алмасады, олар алдымен байланыс сеансын белгілейді. Сессия ашық болып қалады, құрылғы байланысын үзгенге дейін. Қосылуы арасында анықтау және сеанс аяқталғаннан, сеанс сақталуы керек. Сонымен қа-тар, алдын ала үзілетін сеанс бұл қабатта қалпына келтірілуі мүмкін. Бұл қабаты, бұл міндеттерді өңдейді, (ИНГ орындары, ұстау, қалпына кел-тіру, аяқтау). Қабаты 5 жүзеге асыру келесілерді қамтиды NetBIOS, SAP (хаттама Хабарландыру сессия), PPTP (Point Tunneling Protocol) және SOCKS (Secure Socket).

6 қабаты ұсыну деңгейі болып табылады. Бұл қабат осы қосымша-дан хабарламаны аудару үшін жауап береді, ол синтаксис формасында хабарламаны түрлендіреді, ол төменгі қабаттармен талап етіледі. Бұл қабатта, бастапқы хабарламаны ұсынылымы бірыңғай ұсынымға өз-гертіледі. Мысалы, Сипрограммалау тілінде, жолдар арнайы белгімен \ Оаяқталады. Бұл қабатта, кез келген \ 0 белгілер хабарламадан жойылуы мүмкін. Басқа мысал үшін иерархиялық құрылымдалған деректер си-

яқты деректерді XML-ұсыну. Мұндай деректержалпақ форматқақайта құрылуы керек. Шифрлау (шығыс хабарлар үшін) және дешифрлеу (кіріс хабарламалары үшін) жүреді. 6Қабаты жүзегеасыруөзіне келесілерді кіредіSSL (Secure Sockets Layer), TLS (Transport Layer Security) и MIME (Multipurpose Internet Media Extensions).

7 қабаты, ең жоғары қабаты, қосымшаныңқабатын білдіреді. Бұл деңгейде, түпкі пайдаланушы немесе қолданбалы бағдарлама хабарлама жасайды, ол берілуі мүмкін, немесе бұл деңгейде қолданбалы бағдарлама түпкі пайдаланушы үшінқабылданған хабарды көрсетеді. Бұл қабат әр түрлі желілік коммуникациялық бағдарламаларды қамтиды, мысалы Telnet; FTP; электрондық пошта хаттамалары сияқты POP, SMTP, IMAP; және желілік қызметтерді қолдау,домендік атаулардың жүйесі сияқты (DNS).

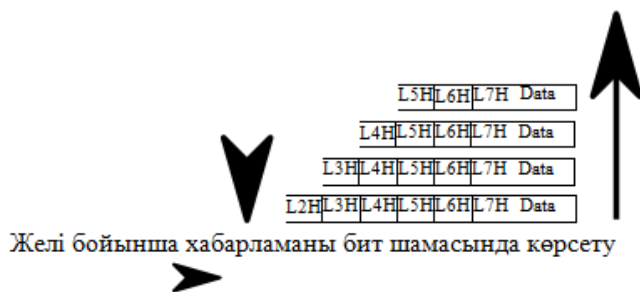
OSI моделікелесітүрдежұмыс істейді. 7 деңгейіндеқандайдабірқосымша арқылыбағдарламалық қамтамасыздандыруменхабарламақұрылады. Бұлбастапқыхабардеректерболыпесептеледі және олберілуі тиіс. 7 қабатыхабардыңалдыңғыбөлігінетақырыптытіркейдіжәне 6қабаты хабарламаны көрсетеді. 6 қабатындағы, ерекшетақырып, сондай-ақхабарламаға қосылады. Хабар, қазіргіуақыттатұратынбастапқыдеректерменекітақырыптарынкөрсетіледіқабаты 5, албасқатақырыпхабарламаныалдыңғыбөлігіндетіркелген. Жоғарғыүшқабаттұтас хабарламада жұмыс істейді, сегментелген блоктарда емес, жұмыс істейтін4 қабатынан бастап. Қабатындағы 4, хабарламапакеттерге бөлінбейді, соданкейіннеғұрлымұсақбірліктерге, әзірге1 деңгейінде, хабарламалар, жекелегенбиттерретіндеқарастырылады. Қабаттар 4 және 3, сондай-ақтақырыптың ақпараынқоядыжәне2 қабаты өзінекелесілердіқосадытақырыбыжәнетөменгісілтемелердің ақпаратын. Және, ақырсоңында, 1 деңгейінде, хабарламаберіледі. Алынғанхабарламажоғарығақарайұқсасбейнеленедіәрбірқабатыарқылы, тақырыптыалыптастау (жәнетөменгі) , әзірге, жәнежоғарғықабатына хабаркеліптүседіжәнеқолданбалыбағдарламалыққамтамасыздандыруғажеткізіледі. Сурет 12.12 иллюстрациясы OSI процесінкөрсетеді.

TCP / IPбасқаболыптабылады, әдетте, деректерберу хаттамасыпайдаланылады. TCP / IP екіжекелегенхаттаманықамтиды, ол Интернеттің бастапқыдамуы кезінде біріктірілген. TCP сенімділіктіжәнереткекелтірілгенпакеттержеткізілуінөңдейді. Олстекхаттамалардыңжоғарыдеңгейлерде жұмыс істейді, бөлшектерді келесілер үшін сақтайотырады, басқадапакеттерді IP бөлігінеберілгенкезінде. Дегенмен TCP Интернетбайланысарқылыкеңіненпайдаланылады, кейбірқосымшаларөзорнындапайдаланушыныңдейтаграммхаттамасын пайдаланады (UDP).

Екі TCP және UDP Интернет-хаттаманың IP үстінен орналасқан. Мұнда біз қысқаша TCP / IP қабаттарын қарастырайық және OSI табылған олардың қабаттарын салыстырамыз. OSI қарағанда, TCP / IP қабаттарының салынбаған. Яғни олар осы қабаттардан еболып жатқанын сипаттайды, бірақ олар осы іс-әрекеттер болуы тиіс егжей-тегжейлі апараты жоқ. Осылайша, көптеген TCP / IP стектің іске асыратын әртүрлі тәсілдері бар. IP ең төменгі қабаты Link Layer деп аталады. Link Layer тікелей екі төменгі деңгейлерге OSI-ға болып келеді. Бұл қабат барлық қызметтерді жеке және берілген байланысқа бағындырады. IP екінші қабаты Интернет Layer болып табылады. Бұл қабат үшін жауап беруге мәліметтер пакеттерінің арқылы бірнеше одаңда көп желілері. Бұл деңгейде, адресі теу және маршруттау жүреді. Дәл осы деңгейде, ол IP-мекен-жайын пайдаланылады. TCP / IP қабаты 2 қабаты OSI 3 қабатына ұқсас болып келеді. Дегенмен, OSI құрылғылар арасында байланысқа мүмкіндік береді, сол желі қабаты 2 болса, барлық маршруттау 2 TCP / IP деңгейінде жүреді.

OSI Layer 3 және TCP / IP арасында басқа дәлелуі айырмашылықтар бар 2 қабатының, бірақ олар бұл жерде қаралмайды.

Айта кету керек, адресі теу және маршруттау, аталған жерде IP нұсқасы 4 (IPv4), ағымдағы және одан астам танымал нұсқасы IP. Дегенмен, шектеулер саны IP-мекен-жайларының, сондай-ақ қызығушылықпен жақсартылады, Интернет, IP-нұсқа 6-қолжетімді болды. IPv6 жасайды дәлелуі айырмашылықтар пайдалану Интернет, себебі сұраныс өседі, келесі он жылдықта. Ең үлкен айырмашылықтарды IPv4 және IPv6 нысаны мекен-жайлары болып табылады. "IPv4, машиналық мекен-жайы 32 бит бөлінген 4 октеттен тұрады. Әрбір октет 8 бит (1 байт) және, осылайша, сандық тауға мүмкіндік диапазоны 0-ден 255. Октет IP-мекен-жайы, нүктелермен бөлінген. Мысалы, IP-мекен-жайы 127.31.49.6 болуы мүмкін



12.12 Фигурасы OSI моделі қабат қабатпен көрсету

Желілердің IP-адресеуге қатысты үш классы бар(ИНГ сыныптарын іс жүзінде асыру желілерін тоқтатылды 1993 жылы болғанымен, желілік мекен-жайлары жиі әлі де берілді бұл жолы). Класс желісін анықтайды, қандай соқалар төрт октета көрсетеді желісі, орналасқан компьютер, ал қалған биттер білдіреді мекен-жайы компьютер желісін). А сыныбының желісі тек алғашқы октеттің 7 бит пайдаланады желілік мекен-жайын белгілеу үшін.

Бұл желі санын А классына дейін шектейді, 128, бірақ мекен-жайлар үшін 25 бит қалдырады және, осылайша, класстағы ғаламтор желісінде 16 млн. (16 млн) ішкі мекен-жайлар болуы мүмкін. В класы желісінің белгіленеді бірінші октета мекен-жайы 128-191, желіні сәйкестендіру үшін алғашқы екі октетті пайдаланады. Бұл 16 бит үшін желілік мекен-жайларын қалдырады немесе 65536. желілерін сынып оқушысы С, ең көп таралған пайдаланады алғашқы үш октета мекен-жайы, мүмкіндік беретін үшін миллион желілерін сынып С. Дегенмен, сынып желісімен, яғни тек бір октет үшін мекен-жайы және әрбір құрылғы. Осылайша, желі С-класты болуы мүмкін ғана дейін 256 мекен-жай мен құрылғылар. Кесте 12.2 бөлшектер үшін. Мынаған назар аударыңыз, бұл екі сынып резервтелген маңызы бар қаланың бірінші октета мекен-жайы, бірақ осы уақықа дейін пайдаланылмаған.

IPv4 ұзындығы 32 битті мекен-жайлары бар. Бұл 232 әр түрлі (әр түрлі) мекен-жайларын қамтамасыз етеді, немесе шамамен 4 млрд. бірегей IP-мекен-жайлары. Бұнын саны көп сияқты көрінгенмен үлкен, шектеу мәселесін туғызды, бұған біз өзіміз қол жеткіздік ол яғни шектен-қолмен терминал құрылғылар (мысалы, ұялы телефондар), Интернетке қосылған. Көптеген сонымен қатар, қол жетімді мекен-жайлар болғанмен де, толығымен пайдаланылмады. Мысалы, сынып А желісінің мекен-жайының жеткілікті саны болғанмен 16 миллионнан астам ішкі құрылғылары бар, бірақ бұл әрбір сынып желісін толық пайдаланады дегенді білдірмейді. Екінші жағынан, ұйым болуы мүмкін желілік адресі С-класты, бірақ қажет болуы мүмкін астам 256 әр түрлі мекен-жайлар.

IPv6 мекен-жайының 128 бит ұзындығы жиі бейнеленеді он алтылық сандық көрсеткішті пайдаланады. Неғұрлым ұзын мекен-жайының артықшылығы ол түрлі 2128 артық IP-мекен-жайларды қамтамасыз етеді. Бұл бізге көп мекен-жайды өте, өте ұзақ уақытта қолдану мүмкіндігін береді, тіпті егер біз басқа мекен-жайы бойынша планетада әрбір процессорды қамтамасыз ететін болсақ-та (соның ішінде ұялы телефондар, датчиктер және басқа электрондық құрылғылар). 128-биттік мекен-жайы, әдетте, екі бөлімнен тұрады, 64-битті желілік мекен-жайлар префикс үшін пайдаланылады, маршруттау, Интернет арқылы, сондай-ақ 64-бит-

тік идентификаторы интерфейс белгілеу үшін хост желі ішінде. Үлгісі болуы мүмкін мекен-жайы 1234: 5678: 90ab: CDEF: 1234: 5678: 90ab: CDEF. Назар аударыңыз пайдалану, он алтылық әріпі", - бинарлық немесе ондық, біз, әдетте, видеофильмдер қараймыз мекен-IPv4. ТаБLe 12.2 Интернет қоры Желілік класстар

Octets for Classwork Address		Octets for Network Address		Number of Network Addresses (1st Octet)	Comments
A	1	3	16,777,216	0–127	Many addresses have gone unused
B	2	2	65,536	128–191	
C	3	1	256	192–223	
D	Not defined	Not defined	de-Not defined	224–239	Multicast addresses only
E	Not defined	Not defined	de-Not defined	240–255	Reserved (future use)

TCP, TCP / IP жоғарғы қабаттар, екі қабаттан тұрады: Transport Layer және қолданбалы деңгейде. Транспортты деңгей транспорттық деңгейге ИОО (OSI Layer 4) ұқсас болып келеді. Алайда TCP хаттамасы екі әртүрлі мәліметтер ағынына мүмкіндік береді, олар TCP хаттамасын пайдаланады және UDP ны пайдаланушылары.

TCP ға қарағанда UDP байланыстың кепілді формасын қамтамасыз етпейді. Беру кезінде UDP пакеті өшірілген кезде, оларды жіберу үшін ешқандай талпыныстар болмайды. Алайда бұл жат естілгенмен де бұл әлдеқайда пайдалырақ болуы мүмкін, нақты уақыттағы араласып тұру әлдеқайда маңыздырақ болып келеді, мәліметтердің сенімді болуына қарағанда. Мысал ретінде нақты уақыт тәртібінде дыбысты жіберу, TCP орнына UDP ны пайдалануға болады. Бұған дәлел ретінде пайдаланушының дыбыс дабылының үзілгенін қаламайды, бұл уақытта пакет қайталанып тұрса да.

Бір (немесе бәрнеше пакеттерді) ескермеу алушының дыбыс дабылын түсінуге ешбір кедергі білдірмейді.

12.13 Суреті TCP пакетінің UDP пакетіне қарсы мазмұнын бейнелейді. UDP пакеті әлдеқайда қысқа болып келеді, өйткені оған ақпарат жетпейді, реттік саны ретінде, яғни ақпараттың шынайылығы, жалауларды басқару, ол пакеттердің қай күйде өңделу қажеттілігін көрсетеді және орын ауыстырылатын мәліметтер, мәліметтер аумағында шамасын көр-

сетеді. TCP пакетінде тез уақыттағы көрсеткіші міндетті болып келмейді, алайда мәліметтердің соңғы байтын белгілеу үшін пайдалы болып келеді.

TCP / IP ең жоғарғы қабаты болып қосымша деңгейі болып саналады, ол шамамен ИОО жоғарғы қабатының синонимы болып келеді. ВОС деңгейінің сессиясында, мұндағы байланыстар

Дереккөз мекені

Белгілеу көзі

Дереккөз порты Белгілеу порты

Сандар реттілігі

Мойындауды білдіру номері

Берілген офсет/Терезе жалауының резервтелген биттері

Бақылаушы суманы көрсету Тез

Варианттар (опиционалды)

Мәліметтер

12.13 Суреті Стандартты TCP (а) пакеті және UDP пакеті (б).

Аяқталғанға дейін сақталады, TCP / IP нақтырақ эквивалентті ештенесі жоқ. Алайда бұл мүмкіндіктер TCP / IP транспорттық деңгейінде өңделеді.

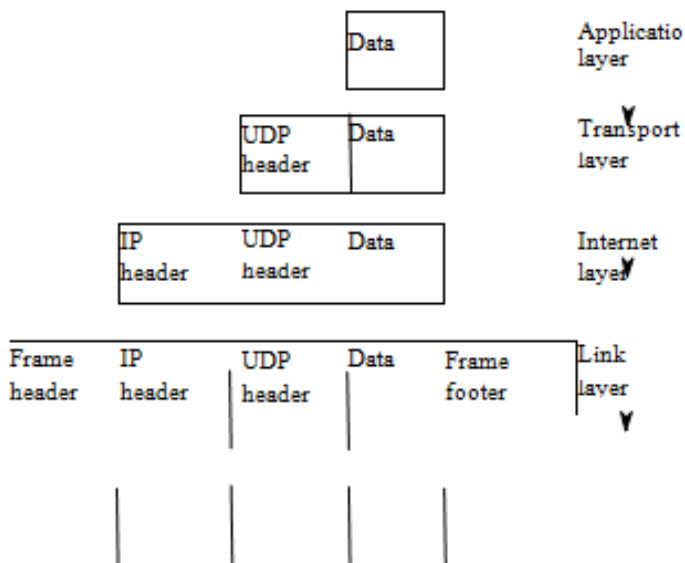
Соңына дейін сақталады, TCP дәл баламалы ештеңе жоқ / IP. Дегенмен, ондай мүмкіндіктер көлік деңгейінде өңделеді. TCP / IP.

Төрт топтарын TCP / IP бірге ұсыну, біз сол жағынан көп бөлігінде, ретінде OSI функционалдығын көреміз. OSI сияқты, хабар немесе алынған қабаттары арасында өзгертіледі. Сурет 12.14 мысалы келесіні береді қабат қабатымен пакеттің айырбасталғандығын көрсетеді TCP / IP, онда әрбір төменгі қабатыалдыңғы қабатына өз тақырыбын қосады және Link Layer қосады, сондай-ақ төменгі колонтитулды қосады.

Естеріңізге сала кетейік, екеуі де OSI және TCP / IP хаттамалары. Олар-байланыс желілерінің кейбір аспектілерін ұсынады, олар өткізушілерге басқа да бөлшектерді қалдырады. Екеуі де олар үшін хаттамалар қолданылды аудару үшін хабарлама бір форматты (сгенерированные кейбір бағдарламалық қамтамасыз етуді қолдану катиона) форматы, ол берілуі мүмкін, содан кейін ауыстырылды кері бюджеттен берілген нысанның пішіні, мақсатты қосымша бағдарламалық қамтамасыз етуді өңдей алады. Оралыңыз суреттер 12.12 және 12.14 көрсету үшін OSI және TCP / IP, тиісінше. Ретінде сіз байқамайды, бейнелеу орын жоғарыдан төмен қарай және төменнен жоғары қарай.

Шамасына қарай, хабарлама жылжиды төмен хаттама бойынша көп бөлшектер қосылады сияқты тақырыптары, қателер тіркеу келгенде

және адресстеу. Хабарлама ретінде жылжиды жоғары хаттаманың бұл қосылған бөлшектер сняли хабарлар. Тағы бір аспект жазбалар үшін TCP / IP болып табылады желі құттықтау. Шын мәнінде, әрқашан екі компьютер деректерімен алмасады, желі қатар жүреді кез келген хабарлама, екі құрылғы орындау керек желілік қол алысу. Шын мәнінде, бұл бірінші машина контактирует екінші-ақ, ол көрсетеді "Менің сізбен қарым-қатынас", содан кейін оны күтеді. Кезде екінші машина дайын, ол: "Мен дайынмын". Кейін квитирующий жасалды, машиналар еркін қатынасқа түсуге, әзірге байланыс жоқ аяқталады ұшы. TCP / IP пайдалануға берілді ұғымы үш жүрісті кранның қолалысуымен. Бұл ретте, бірінші аппараты жібереді пакет синхрондау (SYN), екінші машинаға. Екінші машина, егер ондай бар болса, жауап беруші кері синхрондау және қабылдауды растау мемлекеттік пакетін (SYN / ACK). Және, ақыр соңында, бірінші машина әрекет етеді, екінші растау пакетін қабылдау өз (КСҚ). Сол кезде екі машина синхронды және жұмыс істеуге дайын, әзірге хабар жоқ тоқтатылды. Қайта оралсақ, сурет 12.13, сіз көре аласыз "квитирование нөмірі" топтамасы TCP. Бұл үшін пайдаланылады қолдау "үш жақты құттықтау.



ТСР / ІРпакеттерін жасақтау.

Бздің қарауымызда бір тұжырымдама қарастырылмайды, TCP / IP желіліері порт желісі болып келеді. TCP және UDP хабарламалары жеткізу мекенжайын ғана емес, сондай ақ порт мекенжайын талап етеді. Порт тек хабарламаға арналған болжамдық бағдарламалық жасақтама міндетінің көрсеткіші болып табылады. Осылайша, машина арқылы хабарлама алғанда, операциялық жүйе хабарламаны порттың мекенжайы бойынша қалай өңдеу керектігін анықтай алады. Бұл хабарламаны осыған жауап беретін қолданбалы бағдарламалық қамтамасыз ету емес, TCP / IP қалай бөліп салыну керектігін анықтайды. Порттар әртүрлі мақсаттарға арнаған. Мысалы, электронды поштаның SMTP пайдаланылатын жалпы порт, 25 болып табылады, сондай ақ HTTP үшін веб-парақшаларды қарастыруға арналған жалпы порттар болып 80 және 8080 табылады. Порттардың FTP (20), Telnet (23), SSH (22) арналған нөмірлері, және HTTPS (443) хаттамасы болады.

Порттар мекенжайы болып тек қана ОЖ ғана емес, сондай ақ бағдарламалық қамтамасыз ету қорғаныс үшін пайдаланылуы мүмкін. Мысалы, брандмауэр 23 сияқты порттардың белгілі бір мекенжайларына тыйым салуы мүмкін, (себебі Telnet қауіпсізсіз деп саналмайды). Дәл сол сияқты веб-сервер күткен порттардан емес (мысалы, 80, 8080 және 443), басқа порттардан келген хабарламаларды қабылдамай қоюы мүмкін. Алайда біз порттар туралы тақылауымызды осы жерден аяқтаймыз, сіздер, шүбәсіз, әрі қарай зерттеу үшін олармен жұмыс тәжірибесін аласыз. Біз осы тарауды, таратудың желілік мекенжайымен (NAT) байланысты, бір TCP / IP тақырыбы бойынша аяқтаймыз. NAT бір IP мекенжайының екінші IP мекенжайға түрленуі. Basic NAT жай бір IP мекенжайды екіншісінің лақап аты үшін пайдаланады. Ауыстырудың бұл түрі мекенжайлары сәйкес келмейтін екі желі бір бірімен араласуға әрекет етсе пайдалы болуы мүмкін. Негізгі NAT-та ауыстыру үрдісі хабарламаның IP мекенжайын, IP тақырыбын, сондай ақ хабарламаның мазмұны және тақырып ақпаратының кез келген бақылау сомасын өзгерту керек. Әйткенмен, IP мекенжайының бүтін бір топтамасын жасыру үшін, NAT-ты пайдалану үшін ол аса көп таратылған болып саналады. Мысалы, ұйымдарда өзінің ішкі құрылғылары үшін жеткілікті мөлшерде IP мекенжайлары болмауы мүмкін. Бұл жағдайда жергілікті торапта машиналардың кейбіреулері ғана сырттан «көрінетін» нақты IP мекенжай келтіріледі. Басқа құрылғылар сыртқы көзден жасырын. Хабарламалар түскенде, хат алушының мекенжайы ішкі, дербес IP мекенжайға өзгертілуі керек. Бұл түрі NAT көпті біреуге, немесе NAT қайта жүктеу немесе IP бүркемелеу деп аталады. Сервер сыртқы мекенжайдың орнына қандай дербес IP мекенжай пайдаланылғанын анықтайтын жағдайда бо-

луы керек. Дербес мекенжайға ауыстырылған IP қосымша түрлендіруді талап етеді, себебі ішкі бағдарлағышқа десте дұрыс жүргізілуі керек. Түрлендірудің ішінде, порттар мекенжайы, сондай ақ IP мекенжайлары өзгертілуі мүмкін немесе тіпті ықтимал. Көпті біреуге NAT артықшылығы ол ішкі мекенжайдың құпиялылығын қорғау дәрежесін, қосымша IP мекенжайларын алу үшін жергілікті торапқа рұқсат алуды ұсынады. Көпті біреуге NAT негізгі кемшілігі ол ғаламторға қосылуды өңдейтін сервер арқылы енгізеді.

Осы бөлімде біз желілік бағдарламалық жасақтың жалпы түрлерін қарастырамыз. Бұл бағдарламаны біз желіде құрылғылардың арасында деректер алмасу үшін пайдалана алар едік. Олардың бәрі желішілік TCP / IP пайдаланылады, сондықтан ғаламторды ғана пайдаланумен шектелмей, сондай ақ жергілікті торап шегінде қызмет көрсетуге қол жеткізуге болады.

Telnet – бұл желі бойынша басқа компьютерге кіруге мүмкіндік беретін бағдарлама. Telnet арқылы әрекет ететін адамға қашықтан басқаратын пультінде (хост) компьютердің есеп жазбасы болу керек. Телнеттің бір түрі, Linux колжетімді, Rlogin деп аталады. Rlogin- мен бірге пайдаланушыға шын мәнінде мақсатты компьютерге кіріп керек емес, себебі пайдаланушының есептік жазбасы туралы ақпарат (аты, құпия сөз) желі компьютерлерінің бәрінде бірдей. Linux-тің, шын мәнінде, бірнеше «R-коммуналдық қызметі» бар, мұндағы «R» «қашықтан басқаратын пульт». Қашықтан кіруден басқа, RFile, RSH, rstatus және rwho бар. RFile бағдарламасы файлдарды басқару амалдарын қашықтан басқаруға мүмкіндік береді. RSH бағдарламасы қашықтан басқа машинаға сыртын ашады. rstatus бағдарламасының құрамында қабылдағыштың жай-күйі туралы негізгі ақпарат бар. Rwho бағдарламасы Linux, қашық машинадан, сізге жүйеге кім кіргенін көрсете отырып, бұйрықты орындайды. r-коммуналдық қызмет идеясы болып олардың серверде бір журналды пайдаланып, жергілікті торапта жұмыс істеуі болып табылады. R-утилиттер сіздің қашықтағы компьютерге өз жергілікті компьютеріңізге кірмейінше кіруіңізді талап етпейді. Алайда, r-коммуналдық қызметтерді жүйелік әкімшінің пайдалануы үшін айқын орналастыру керек. Telnet және Rlogin мәселенің бір бөлігі ғана; бір де бір бағдарлама қауіпсіз емес. Жергілікті торап пен қашықтағы компьютер арқылы жіберілген мәтін үлгідегі хабарламалар. Telnet жағдайында жергілікті тораптан қашықтағы компьютерге жіберілген қандай да бір құпия сөзден тұрады. Сондықтан хабарламалар құпия сөзді және басқа сезімтал материалдарды көріп қоятын үшінші бір жақпен таңдалуы мүмкін. SSH пайдасы, Linux, автоматты түрде кез келген хабарламаны шифрлаумен

және шифрды ашумен айналысатын, Telnet және Rlogin-ге қарағанда, сонысымен хабарласудың қауіпсіздігін қамтамасыз ететін бағдарлама. SSH Linux-тің бөлігі болса да, ол Windows үшін бояу бағдарламасы арқылы да қолжетімді.

Морриса Ғаламтор WORM

1988 жылдың 2 қарашасында Корнельск университетінің аспиранты Роберт Моррис ғаламторда, Unix қауіпсіздігінде тесіктерді көрсетуге арналған, бағдарламаға ерік берді. Қазірде Моррис «Ғаламтор-құрт» атымен әйгілі бағдарлама 3 күннің ішінде Unix-тің 6000 машинасына енді, нәтижесінде ғаламтордың айтарлықтай бөлігі жорғалауға, немесе толық төмен (бағалауға сәйкес ғаламтордағы бар компьютердің 60 000 жуығы, сондықтан құрт әсері ғаламторда - 10%). Құрт арнайы Unix жүйесіне төрт тәсілдің көмегімен кіруге тырысты:

- Әлсіз құпия сөздерді табу (соның ішінде жалпыға арналған құпия сөз)
 - Sendmail—да Unix бағдарламасының, толып қалған буфер сияқты, белгілі кемшіліктерін пайдалана отырып
 - Unix сұққысы командасындағы белгілі кемшіліктерді пайдалана отырып
 - Желідегі басқа компьютерлерге тарату үшін R-утилиттерді пайдалану
- Құрт машинаға жеткен соң ол өзінің толық бағдарлама нұсқасын жүктейді және Желідегі басқа машиналарға, кестеде аталынған машиналар/ж.т.б./ағымдағы машина түйіні, және басқа машиналарға жоғарыда көрсетілген, төрт тәсілмен жіберу үшін көшірмелер жасайды.

Шындығында, құрт қызмет көрсету бағдарламасынан бас тарту қызметін атқарды, ол өзіне көптеген көшірмелер жасағандығы сондай, зақымданған компьютерлер құрттың көшірмесін іске қоспас бұрын әлі де біраз жұмыс істей алар еді.

Жұмсаған күш-жігері үшін Моррис \$10000 айыппұл төледі және 400 сағат қоғамдық жұмыс істеді. Құрттың келтірген залалы шамамен \$100000 и \$10 млн. бағаланады. Құрт болашақта қатер және төтенше жағдайлармен айналысатын топтарды құру үшін DARPA сұранысы (алдыңғы қатарлы әскери жобаларды зерттеу агенттігі).

Пинг, түрлендіретін, және дестелерді қайталап тағайындау құрылғысына жеткізетін, расталған хабарламаларды шығарумен айналысатын бағдарлама. Бұл қашықтағы компьютер қолжетімді болса, сіздің машинаңыз (2) желіге қол жеткізе алатындығыңызды білуге мүмкіндік береді (1). Сіз сондай ақ желіде қандай да бір ерекше кешігулер бар ма, хабарламаның тез немесе баяу қабылданатынын көру үшін пингті пайдалана аласыз. Пинг үшін ұқсас бағдарлама ол жол тарту. Бұл бағдарлама сіздің

компьютеріңіз бен бекітілген компьютер арасындағы бағытын шығарады. Сіз жол тартуды хабарламаның оны ғаламтор арқылы қалай жасайтынын, жолда мәселе тудыратын бағдарлауыштар болса, қанша уақытта қабылдайтынын көру үшін пайдалана аласыз. Пингтің екеуі де және жол тартулар көбінесе қауіпсіз бағдарлама болып есептелінеді. Ақылды хакерлер жергілікті торапта компьютерді зерттеу үшін осы бағдарламаның біреуін немесе екеуін де пайдалана алады. Сөйтіп, хакерлер жұртшылықтың игілігі бола бермейтін IP-мекенжайын қолға түсіреді, және осы IP-мекенжайларды шабуыл ұйымдастыру үшін қолданады. Сондықтан, желілік әкімші немесе жүйе екі бағдарламаны да өшіріп тастауы әбден мүмкін.

HTTP қатынастың веб-шолғышы үшін негіз болып саналады. HTTP сұранысы веб-серверге жіберіледі, паракқа әкеледі, кері қайтарылады және шолғышта бейнеленеді. Алайда, егер сіз файлды шолғышта бейнеленбейтіндей етіп тез жүктегіңіз келетін жағдайлар болады. Мұны сіз Linux – те командалық жолдан Wget бағдарламасының көмегімен жасай аласыз. Wget-те толық URL-мекенжайыңызды көрсетуіңіз керек (сервердің атауы, сервердегі каталогке жол, файл атауы). Содан кейін бұл файл ағымдағы каталогке жүктеледі.

Сіз сондай ақ файлды FTP-бағдарламасының көмегімен тікелей жүктей аласыз. Дегенмен FTP-де сіз бен сервердің арасында сеанс аяқтамайынша немесе үзіліс алғанша ашық тұратын сеанс пайда болады. FTP-бағдарламасының басым бөлігі графикалық болып келеді, сондай ақ командалық жолдың нұсқасы да бар. FTP-дің графикалық нұсқасының құрамына FileZilla, Win-Ftp және Win-SCP кіреді (SSH Telnet –тің қауіпсіз түрі болғандай, құрамында FTP-дің қауіпсіз түрі бар). FTP-ті қашықтықтағы компьютерде файлдарға қол жеткізу үшін қолданғанда сіз желіге анонимді қолданушы ретінде кіруіңіз керек. Анонимді қолданушылар FTPсерверінде ашық файлдарға қол жеткізе алады.

Үлкен бір айырмашылығы HTTP және FTP арасында URL айырмашылық болып табылады. Сіз әдетте HTTP сұранысты веб-шолғыштағы HTML-құжат сілтемесіне басып жібересіз. Сілтеме сіз алғыңыз келген құжаттың тұрған орнына код қояды, сондықтан мұндай ақпаратты есте сақтап керек жоқ. FTP бағдарламасында алайда, сіз сервердің мекенжайын білуіңіз керек, содан соң не каталогқа жолды және файлдың атын білуіңіз керек немесе сервердің айналасынан сізді қызықтыратын затты табу іздеуіңіз керек.

HTTP және FTP сервер мекенжайы құрылғының IP-мекенжайы болу керек. Алайда IP-мекенжайын есте сақтау қиын болғандықтан, оның орнына біз IP-аппараттың лақап атын пайдалана алатын құлақша ойлап таптық. Мақсаттағы машиналардың

сипаттамасы (немесе аббревиатура) сияқты лақап ат ағылшын тілінде. Мысалы бізде кәсіпорындар үшін www.google.com , университеттер үшін www.nku.edu және үкіметтік сайттар үшін www.usa.gov сияқты лақап аттар бар.

Бағдарлаушылар IP-лақап аттарынан емес, тек қана IP-мекенжайымен ғана жұмыс істей алады, сондықтан біз трансляция түрінің лақап ат мекенжайынан жүргізілуін қамтамасыз етеміз. Бұл DNS-пен өңделеді. Сервер DNS тек қана компьютер болып табылады, бейнелеу ақпаратын ұсынатын файл деректері мен DNS бағдарламасы жұмыс істейді. Бұдан басқа DNS ғаламторда бір немесе бірнеше DNS сілтемесі бар, ол сұранысты басқа DNS беруі мүмкін. DNS ғаламтор арқылы иерархиялық ретте орналасқан, осылайша сұраныс бір DNS бірге басқаға, басқадан басқаға, бейнелеу аяғында алынуы мүмкін.

Nslookup бағдарламасы IP-лақап аттан IP-мекенжайларды сіз үшін көшіруді қамтамасыз етеді. Егер сіз нақты лақап ат DNS кестесінде бекітілгендігіне көз жеткізгіңіз келсе бұл өте ыңғайлы. Егер сіз оның лақап атын ғана біліп, компьютердің IP мекенжайын білгіңіз келсе бұл сондай ақ пайдалы.

Егер сіз компьютеріңіздің IP мекенжайын білгіңіз келсе, Ifconfig (в Linux) немесе IPCONFIG (в Windows) бағдарламаларын пайдаланыңыз.

Агралар Linux желісінің басқа үш командасы болып табылады, Ethernet және желілік картаны желіге қосылуын анықтау үшін қолданылады; NetStat желіге қосылуын және сокеттің жағдайын анықтау үшін пайдаланылады; бағдарлау кестесі бейнеленген бағытты (жұмыс шлюзінің IP мекенжайы желісі және жергілікті тораптың IP мекенжайы сияқтыларды қоса алғанда). Сондай ақ Linux желі қызметінің командасы екендігіне аса көңіл аудару керек. Сіз желі қызметін жүргізе немесе тоқтата аласыз немесе оның жағдайын тексере аласыз. Команда келесі сипатта: / SBIN / командалық желінің қызметі

Төмен, жоғары, іске қосу, қайта іске қосу, мәртебе бұйрықтары бар жерде. Желі қызметі ол желіні пайдалану мүмкіндігіңіздің болуы үшін (хабарлама жіберу немесе алу үшін) сіз үшін іске қосылған демон. Linux желімен байланысты, зерттеуге тұрарлық, көп файлға ие.олардың кейбіреулері төменде келтірілген:

- / ж.т.б. / түйіндер: сіздің машинаңыз жиі тілдесу үшін, машиналар үшін мекенжайларды бейнелеу үшін IP-лақап аттар. Осы файлға IP- мекенжайларды жүктей отырып, сіздің машинаңыз DNS байланыспауы керек, біріншіден уақытты үнемдейді.
- /etc/resolv.conf: сіздің DNS мекенжайын сақтайды. Онда бірнеше жазбалар болуы мүмкін.

- / ж.т.б. / sysconfig / скрипт-желілер /: желілік қосылуды ретке келтіру үшін Linux жүйе инициализациясы үдерісінде іске қосылған конфигурациялық файлдар сақталады
- /etc/xinetd.d/: желілік демонға сенім артқан қызметтер, XINETD.
- /etc/hosts.allow и / ж.т.б. / hosts.deny: сіздің компьютеріңіздің қолжетімділігіне рұқсат беру немесе шектеу қою.
- /etc/hosts.equiv: «сенім білдірген» машиналардың IP мекенжайлары бар, сондықтан балама ретінде анықталған r-коммуналдық шаруашылық барлық компьютерлерде жұмыс істейтін болады.

Сіздің компьютеріңіздің желіде орналасуы хакерлерді сіздің компьютеріңізге шабуыл жасауға шақырады. Бұл заманауи есептегіш техникалардың ең осал тұсы шығар. Компьютеріңізді желіге қоспас үшін, ақырғы қауіпсіздік. Әрине, бұл компьютердің мүмкіндігіне күрт шектейді. Осылайша желілік қауіпсіздіктің бағдарламалық жасақтамасы қолжетімді. Осындай бағдарламалық жасақтама тек қана пайдалы ғана емес, қазіргі кезде ғаламторға қол жеткізгісі келетіндер үшін де қауіпсіз.

Кең тараған бағдарламалық жасақтама қауіпсіздігінің екеуі брандмауэрлер және вирусқа қарсы бағдарламалық жасақтама болып табылады. Брандмауэр бағдарламалық немесе аппараттық жасақтама болып келеді (Сонғы жағдайда бұл Брандмауэр бағдарламалық жасақтамасымен жұмыс істейтін, ерекшеленген сервер).

Брандмауэр бағдарламалық жасақтамасы құрамында брандмауэр және компьютер арқылы жасауға рұқсат берілген, және тосқауыл қойылуы тиіс, хабарлама түрлерін сипаттайтын ережелер тобы бар. Ережелер портқа, бірқатар хабарламаларға, хабарлама түрлеріне, дереккөздің IP мекенжайына, хабарлама TCP немесе UDP форматта ма, және хаттамаларға (мысалы, HTTP, HTTPS, FTP) негізделген.

Біз 11 тарауда брандмауэр Linux қысқа қарастырып шықтық. Вирусқа қарсы бағдарлама егер файлда вирус болса, немесе БЖ зиян келтіретін кейбір түрлері болса вирусқа қарсы бағдарлама талап бойынша іске қосылады, немесе сіз оны түскен хабарламаларды дерлік тексеріп отыратындай етіп реттеп алуыңызға болады. Өкінішке орай, вирусқа қарсы бағдарламалық жасақтамалар зиян келтіретін бағдарламалық жасақтаманың әр бөлігін байқай бермейді.

Бағдарламашылар қаперсіз қолданушыларға шабуыл жасауда ойлап тапқыш болып кетеді, вирусқа қарсы бағдарламалық жасақтама күрделене түсуі керек. Алайда бағдарламаға қарсы жасақтама әрқашан шабуыл жасаушылардың жаңалық енгізілуінен артта қалып отырады. Сонымен қатар сіздер жиі вирусқа қарсы бағдарламалық жасақтаманың жаңартылуына ыждағатты болуларыңыз керек (жаңа БЖ күнде немесе апта

сайын шығып тұрады).

Желілерде бағдарламалық жасақтаманың интрузивін және басып кіруді анықтайтын, Сьон алдын алу сияқты өздерінің өзіндік қауіпсіздік түрлері болуы мүмкін. Онда барлық операциялық жүйелерінің қауіпсіздігінде тесіктер болғаны сияқты, сол сәтте басып кірудің алдын алу үшін жалпы шешім жоқ. Шындығында, хакерлер шабуыл жасамас бұрын, қорғанысты тексеріп көруге тырысады. Хакерлер тапқан кейбір тесіктер, брандмауэрді қоса алғанда (мысалы порттар тексерілмейді), қорғансыз, сенімді құпия сөзді қажет етпейтін, жоғарыда келтірілген шифрға сымсыз қосылу, және жол тарту, пинг сияқты кемшіліктер қалдырады.

Бүгін табылған кең тараған шабуыл түрлері веб-серверлерге қызмет көрсетуден бас тарту, хабарламаларды алып қоюдағы IP-спуфинг, шабуылшының машинасы болу үшін бағдарлауыштағы IP мекенжайды өзгерту үшін ARP уландыру, хакерлерге компьютер жадысына өздерінің инъекция кодымен және SQL, хакерлердің деректер қорына өздерінің SQL командаларымен енуі үшін буферді асыра толтыру. Аталған шабуылдардың атақты түрлері болса да, жүйе олардан қорғалған деуге болмайды. Және, белгілі болғандай бұл шабуылдар қазіргі күнде жаңа шабуылдар деп ойлағанбыз. Кез келген деректердің біртұтастығын, өздерінің компьютер жүйесінің функционалдығын бағалайтын ұйым операциялық жүйені және желілік қауіпсіздікті талап етеді. Бұл желілік қауіпсіздік әкімшілерін жалдайтын орта және ірі ұйымдарға тән, қосымша желілік әкімшілерге тән.

Біз деректердің біртұтастығын және қауіпсіздігін 15 тарауда қарастырамыз.

Ғаламтор.

IP мекенжайлары есте сақтау үшін өте күрделі, сондықтан қолданушылар www.google.com. Сияқты лақап атты желеуретеді. Лақап атты нақты IP мекенжайға өзгерту үшін сізге DNS серверіне қол жеткізуіңіз керек. Әрбір желіде DNS бар немесе DNS қайдан табуға болатынын біледі. Қолданушы ретінде сізден көп бөлік жасырылады, алайда егер сіз желі әкімшісі болсаңыз, онда сіз DNS өзіңіз ретке келтіруіңіз қажет немесе оның қайда екенін біуіңіз керек. Қолданушының есіне түсіруі керекті шектеу үшін қосымша лақап аттар болады. Мысалы, ол веб-сервердің келесі лақап аттары үшін жалпы болып келеді, сондықтан қолданушыға WWW енгізудің қажеті жоқ, яғни, www.google.com веб-іздеуіш Google серверінің лақап аты болса да, google.com аламыз. DNS IP мекенжайлары, лақап аттары бар кестеден тұратын компьютер. Ғаламторлық лақап аттарын қолданғанда DNS өз кестең арқылы енуді тексереді содан соң лайықты IP мекенжайды қайтарып береді. Содан кейін IP мекенжай ком-

пьютерге кері жіберіледі.

Компьютер желілері ондаған жылдар бойы өмір сүріп келеді. Өткен уақытта компьютерлік желінің көпшілігі жергілікті тораптық болған (LAN), олар оқшауланған, біз желінің ішінде ғана қатынаста бола алатынбыз. Мұның бәрін ғаламтор өзгертті. Ғаламтор бойынша қарым-қатынас жасау үшін, машина TCP / IP хаттамасын іске қосуға тиіс болатын. Ғаламтордың өсіп танымал болуына байланысты барлық компьютерлер бүгін TCP / IP хаттамасын іске қосады. Жергілікті желіге қосылған ғаламтор, сондай-ақ егер жергілікті желі ғаламторға қосылған болса ғаламтор арқылы біріктіріледі. Біз бір желі аяқталып, екінші желі басталатын сызықты өшіріп тастадық. Бір бөлмеде компьютерлерді басқа бөлмелердегі компьютерлермен байланыстыратын желі бағдарлауышқа жалғанған. Бір қабатта бар желі, басқа қабаттағы желіге қосылған. Бір ғимараттағы желі екінші ғимараттағы желілерге қосылған.

LAN бүкіл университет қалашығын қамтитын осы желіліердің кез келгені ретінде қарастырылуы мүмкін. Қашықтан басқа сайттардан кіргендерді қосу үшін VPN көмегімен желі кеңейтілуі мүмкін. Ұйым желісі де сондай ақ ғаламторға қосылып тұруы мүмкін. Ғаламтор провайдерінің арқасында үйдегі компьютерлер сондай-ақ ғаламторға қосылып тұр. Сондықтан, сіз ғаламторды үйдегі компьютердің LAN желіісінің кеңейтілуі деп қабылдауыңызға болады.

Компьютер деректерін "желімдеудің" үш әдісі бар:

1. Ақпараттарды коммутациялау. Телефон желісі арналар коммутациясын пайдалана отырып сіз телефон нөмірін тергенде, телефонды ажыратқанға дейін телефон шалушы мен телефонда алушы арасында жол орнатылады, және жол айқындайтын және тиянақы болып қалады.

2. Дауыстық дыбыстарды таратқыш (деректерді емес), ұялы телефон желілері де осындай, дегенмен сызбасы жылжытуына қарай өзгертілуі мүмкін, өйткені жіберіледі жақын қол жетімді ұялы телефон мұнаралары жіберіледі. Ақпараттарды коммутациялау, екінші жағынан, шағын деректер дестелеріне бөліп тастайды, және олардың әрқайсысы желі бойынша жеке беріледі. Әр десте баратын жеріне жету үшін басқа жолды таңдауы мүмкін. Дестелік коммутация алғаш рет 1960-жылдары, ғаламтор алғаш рет әзірленгенде пайдаланылды.

3. Бағдарлауыштар. Бағдарлауыштар хабарлама желісі бойынша жедел жіберетін құрылғы. Бағдарлауыштар хабарламаны қабылдайды, мақсаттағы IP-мекен-жайын талдайды, хабарламаны желінің мекені бойынша, басқа бағдарлауышқа жібереді. Бағдарлауыштар мақсаттағы желіде LAN жеткенге дейін хабарламаның жолын жасайды, және, ақырында жергілікті торап арқылы соңғы компьютерге хабарлама жібереді.

4. IP-мекенжайлары. TCP / IP хаттамасының бөлігі адреcтеу қағидалары қалыптастырады. IP-мекенжай төрт саннан тұрады, әрбір сан 0 ден 225 дейін көлемде. Мысалы: 10.11.241.105 IP-мекенжайы болып келеді. Әрбір сан 1 байт жадыда сақталады (1 байт 8 битті құрайды, ал 8 бит кез келген 0 мен 255 ке дейінгі сандардың кез келгенін сақтай алады). Осылайша IP-мекенжайының көлемі 4 байт немесе 32 бит. IP-мекен-жайын есте сақтау қиын болғандықтан, ағылшын тілді атауы болғандықтан, біз жиі IP-лақап атты желеуретеміз (мысалы www.google.com). IP-лақап атты IP-мекенжайға көшіру үшін біз DNS пайдаланамыз. Бұрын айтылғандай, 32-битті мекен-жайлардағы шектеулерге байланысты, IPv6 IP-мекен-жайлары үшін 128 бит пайдаланады

Қазіргі кезде пайдаланылатын IP-мекенжайлардың отырғандықтан 4 октетті мекенжай түсінігін 3 класы мойындатып отыр. IP-мекенжайларының А класындағы бірінші октет желінің сәйкестендіргіші болып табылады, ал қалған бөлігі компьютер нөмірін құрайды. В класы мекен-жайы бөлуге ID мекен-жайы желілері алғашқы екі октеттен және машиналар саны соңғы екі октеттен. С мекен-жай класы машиналар саны үшін (256) бір октет қалдыра отырып алғашқы үш октетаны желілік мекен-жай үшін пайдаланады. Желілік бөліктің мекенжайы класын анықтауда бөлу процесі арқылы деп аталатын, желі астын қалыптастыруға болады. Желі асты бетпердесі деген терминмен Сіз мүмкін таныс шығарсыз. Желі асты бетпердесі дегеніміз не? Біз жиі компьютерде мәндерді AND немесе OR көмегімен екілік сандармен разряды бойынша есептеп шығарамыз. Мысалы екі байт 10101010 және 00001111, екеуі бірігіп AND операциясы 00001010 береді және OR операциясы 10101111 береді. Бұл операциялар битті түрде жұмыс істейді, яғни, біз операция AND немесе OR бірінші бит екі сандарды алу үшін жаңа бірінші бит (1 және 0 тең, 0, 1 немесе 0 1). Біз биттің барлық жұптарымен осылай істейміз. AND операциясы егер екі биттің екеуі 1 тең болса, 1 қайтарып береді (0, кері жағдайда), 1 бит болса да (0, кері жағдайда) OR операциясы 1 қайтарылып беріледі. Яғни, біз оларды неге пайдаланамыз? Сізге келесі екілік разрядтағы IP-мекен-жайыңыз бар деп елестетіп көріңізші: 00001010.00001011.11110001.01101001 (10.11.241.105)

Егер AND бұл 11111111.11111111.11111111.00000000, сонда біз не аламыз? Біз аламыз

00001010.00001011.11110001.00000000

Неге біз оларды пайдаланамыз?

И так, почему мы используем их? Сізге келесі екілік разрядтағы IP-ме-

кен-жайыңыз бар деп елестетіп көріңізші:

00001010.00001011.11110001.01101001 (10.11.241.105)

Егер AND бұл 11111111.11111111.11111111.00000000, сонда біз не аламыз? Біз аламыз

00001010.00001011.11110001.00000000

Бұл мән нені білдіреді? Біз 10.11.241.105 сандарынан бастаған едік және AND операциясының нәтижесі бізге 10.11.241.0 берді. С класы желісі үшін желінің біздің компьютер орналасқан IP-мекенжайын алдық. Екінші жағынан AND 00000000.00000000.00000000.11111111 болса біз аламыз

00000000.00000000.00000000.01101001

Бұл мән, 0.0.0.105 біздің компьютердің желілік нақты мекен-жайы болып табылады.

00000000 және 11111111 Битті моделі 0-ден 255 дейінгі бүтін сандар болып табылады. Біз 255.255.255.0 және 255.0.0.0 мәндерін желілік беттерде ретінде пайдаланамыз. Біз пайдаланбайтын екі комбинация 0.0.0.0 және 255.255.255.255, себебі мұндай бетперделер желіге 0.0.0.0 немесе барлық бастапқы IP-мекен-жайды қайтарады және бізге ешқандай пайдалы ақпарат әкелмейді. Желіасты бетперделер желі нөмірін жылдам алу үшін қолданылады, және желіасты нөмірі құрылғының мекенжайын алу үшін пайдаланылады. Сіз сондай ақ IP-мекенжайды мына түрде көре аласыз: 10.11.241.0/24 немесе 10.0.0.0/8. Бұл желіасты бетпердесін анықтаудың тағы бір түрі. Кейінгі сандар/ 1 бетпердгі екілік қатарына жатады. Осылайша, 255.255.0.0, сондай-ақ 16 ретінде аталуы мүмкін.

Ғаламтор қалай жұмыс істейді

Қысқаша, ғаламтор келесідегідей жұмыс істейді. Сіз компьютерден басқа компьютерге хабарлама жібергіңіз келді (мысалы эл. Хабарлама, веб-паркшаға арналған HTTP сұранысы, FTP, SSH, Telnet сұранысы). Сіздің компьютеріңіз дестеге немесе бірнеше дестеге хабарлама қабылдайды. Егер хабарлама қысқа болса, онда ол бір дестеге сиып кетеді. Мысал ретінде, <http://www.google.com/> сұранысы қысқа болады, бірақ эл. Пошта арқылы ол бірнеше миллион байттық ұзын болуы мүмкін, сондықтан ол бірнеше дестеге сиғызылады. Әр десте алушының мекенін алады (IP лақап ат), (ақпараттық деректе анықталған қателерді бірнеше байттар мәнін қосу арқылы есептеп шығарады, содан соң шыққан санды сығады, содан соң шыққан санға шыққан санды қосады) және кері мекен-жай (соңғы компьютер жауапты жіберу үшін сіздің компьютеріңіздің IP-мекен-жайы). Десте жиналған соң IP бүркеншік атының дестинация IP-мекен-жайына ауыстырылады. сіздің сайтыңыздың DNS мекенін компьютер сақтайды (Linux, /etc/resolv.conf). Компьютер DNS

сайттың хабарламасын IP бүркеншік атты, IP-мекенжайға аудару үшін жібереді.

Егер локальді DNS IP бүркеншік атты танымай жатса, онда сұрау басқа DNS беріледі. DNS Интернетте иерархиясы бойынша орналасқан. Сайып келгенде, DNS лақап аты танытын болады, және мекенжайы бар оларға жауап беретін болады.Әрбір DNS қабылдаушы өз кестелер болады жаңартып отырады (қажет болған жағдайда). Дегенмен, егер DNS иерархия жоғары деңгейінде лақап атты танымайды, жіберілген хабарламаны кері қайтарып жібереді, сіздің хабарламаңыз сіздің компьютерден кетпей жатады.IP-мекен-жайы ұсынылғаннан кейін дестелер жіберіледі. Олар сіздің компьютеріңізден сіздің сайтыңыздың ғаламтор-нүктесіне жіберіледі.Хабарлама ағымдағы жергілікті тораптан шлюзге жіберіледі.Ол жерден сервер әрбір дестені ғаламторға қосылған оның жолдары арқылы жөнелтеді. Әрбір десте өз жолын бір бағыттан екінші бағытқа көше отырып ғаламтор арқылы табады. Әрбір бағдарлауыш ағымдағы хабарлама кестесіне әр түрлі жолмен, қол жеткізуге негізделген жолды таңдайды.Десте бір сілтеме, екі сілтемелер, немесе бірнеше сілтемелер арқылы өтуі мүмкін. Аталған хабарлама үшін қанша транзиттік шлюздердің керек екендігін анықтау үшін зерттеу бұйрығын пайдаланыңыз. Хабарламаны алғаннан кейін жеткізу пунктінде, тағайындалу IP-мекен-жайы нақты хосттың мақсаттағы IP-мекен-жай болуы мүмкін немесе мүмкін емес.

Егер NAT қашықтағы торапта пайдаланылса, онда IP-мекен-жайы "сыртқы жария" мекенжайынан"ішкі жеке" мекен-жайға аударылады.Әдетте, ел ішіндегі өткізгіш жолдарын аз және шын мәнінде хабарлама тек бір бағыт болуы мүмкін.Кейін, түпкі компьютер дестені алғаннан соң компьютердің операциялық жүйесі оны сақтайды.Топтама хаттамалар әйнегінде көрсетіледі, бірдей хабарламалар түсуіне қарай бір жерге топтастырылады.Олар реттік санына қарай сұрыпталады (мысалы, 5 ішіндегі 3). Барлық дестелерді алғаннан кейін соңғы хабарлама тиісті қолданбалы бағдарламалық жасақтамаға ұсынылады (немесе FTP немесе Wget дискісінде сақталады).Назар аударыңыз, дестелерді ретке келтіруді , IP жүзеге асырмайды, неғұрлым жоғары деңгейдегі TCP хаттамасы жүзеге асырылады. Мақсаттағы дестелер хабарламалар толық көлемде және дұрыс алынды деген хабар жібереді (дестелер алынып тасталған жағдайда, немесе қателерімен келіп түссе)

Пакеттер хабарлама жібереді, бұл хабар алынды, және дұрыс (немесе жағдайлар болған жағдайда не пакеттер болды ағызылады немесе пайда қателер).Бүкіл процесі бірнеше секунд немесе одан аз (әдетте), тіпті 1000 миль қашықтыққа қарамастан.

ҒАЛАМТОР БҮКІЛӘЛЕМДІК ТОРҒА ҚАРСЫ

Дегенмен адамдардың көпшілігі терминдер алмасыра отырып қолданса да олар әр түрлі. Ғаламтор осы тарауда - бұл компьютерлік желі. Бүкіләлемдік желі, бір-бірімізбен қарым-қатынас жасау үшін компьютерлер мен компьютерлік ресурстар желісі, барлық құралдарды біріктіретін медиа (ол кабель, радио, микротолқынды немесе спутниктік қосылыстар) және свитч және бағдарлауыштар сияқты хабар тарату құрылғысы. Ғаламтор - бұл физикалық объект; ол қолға ұстап көруге болатын бөліктерден тұрады. Бүкіләлемдік тор – бұл –гиперсілтемелермен біріккен құжаттар жинағы. Бұл құжаттар біз серверлер деп атайтын компьютерлерде сақталады. Бұл серверлер барлық компьютерлер арасында бір ғаламторға біріктіріледі. Бізде ғаламтор web-сіз болуы мүмкін, бірақ ғаламторсыз ешқандай тор жоқ. тағы бір тармақты атап өту керек, адамдар жиі олар ғаламторды жоғалтып алдық немесе ғаламтор жоқ деп түсіндіреді. Шын мәнінде, олар ғаламторға қосылу мүмкіндігін немесе ғаламтор арқылы қарым қатынас жасау мүмкіндігін жоғалтып алғандар. Ол ядролық соғысты, көптеген қираулар болса да оған ештеңе болмайтындай етіп әзірленді

Ғаламтордың қысқа тарихы

1968 жылы төрт ғылыми-зерттеу ұйымдары АҚШ қорғаныс Министрлігімен (DOD) алыс қашықтықта байланыс желілерін құру үшін қаржыландырылды. Бұл ұйымдар Юта штатының Университетінде болды, Калифорния университеті, Лос-Анджелес (UCLA), Калифорния университеті Санта-Барбара және Стэнфордском зерттеу институтында (SRI). Олардың ниеті деректерді, файлдарды қашықтан басқара отырып бір-бірінің компьютеріне қол жеткізуде "электронды миыды" салу болды. Олар салыстырмалы түрде жақында ұсынылған коммутациялау технологиясын іс-тәжірибеде пайдалану жөніндегі идеяны, және осы компьютерлерді телефон желісін қосу идеясын жақсартты. Бұл желі ARPANET деп аталады. Бірінші хабарлама UCLA және SRI арасында 29 қазан 1969 жылы жіберілді. Пайдаланылығыан хаттама 1822 хаттама деп аталатын. Бұл хаттама аса тиімді болмады, сайып келгенде, TCP / IP 1983ж ауыстырылды. Бірінші хабарлама электрондық пошта арқылы 1973 жылы ARPANET 1971 ж. FTP арқылы жіберілді

Заман ағымымен желі жүйесіне көптеген компьютерлер қосыла бастады. 1970 жылдың маусым айына қарай желіде тоғыз компьютер саналынады. 1970 жылдың желтоқсан айына қарай 13 компьютер, ал 1971 жылдың қыркүйек айына қарай 18 саналынады. 1972 бастап 1973 жылдар аралығында 29 содан кейін 40 компьютер ARPANET-ке орналастырылды.

ARPANET –ті қолдайтын екі серік көзделді және желі жүйесін халықаралық ету үшін Норвегия мен Гавай аралдарының екеуіне де 1973 жылы компьютер қосылды. 1975 бастап желі 57 компьютерге жетті. Сонымен қоса, 1975 жылы Қорғаныс Министрлігі ARPANET-тен ARPA-ға дейінгі бақылауды Қорғаныс Байланыс Агенттігіне (DOD) берді. 1983 жылы әскери бөлім желісі (MILNET) бөлініп кетті. Бұл бөлінуге дейін 113 компьютер, ал бөлінгеннен кейін 68 компьютер болды.

1974 жылы Ұлыбритания байланыс пакетімен өзінің жеке желі нысанын дамытып шығарды, сонда X.25. X.25 қолжетімді болды. Басқа желілер 1970 жылдың аяғы мен 1980 жылдың басында пайда болынды. Олар ARPANET-пен қатар, соңында Интернеттің компоненттары болып қалды.

1979 жылы Дьюк-тің екі студенті электрондық пошта арқылы хабарлама алмастыруға арналған UUCP ойлап тапты. Бұл Usenet жаңалығы ретінде белгілі болды. Usenet үйдегі компьютерге немесе желілік құрылғыдан модем арқылы желіге қатынаса алады және бірнеше желіліер мен адамдар үшін қол жетімді болды.

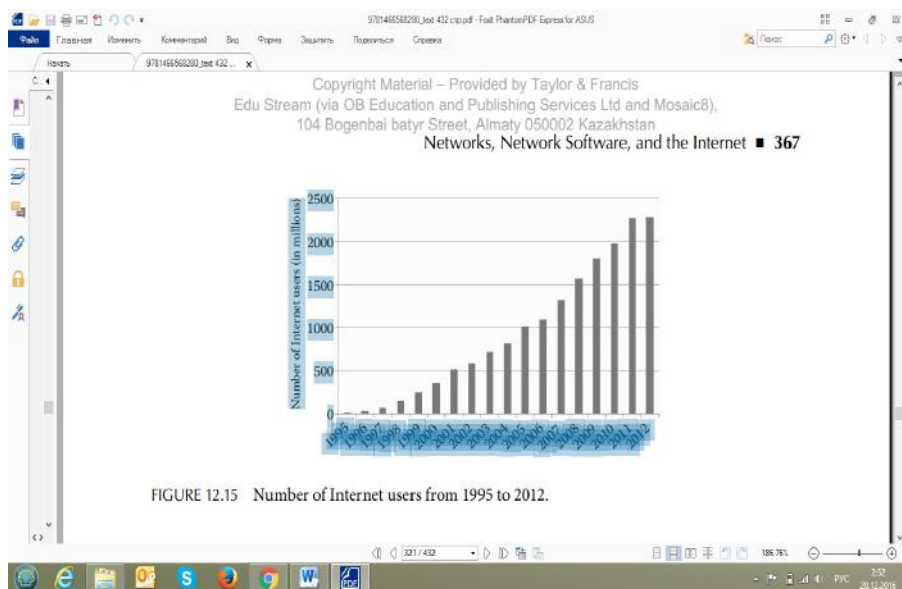
1983 жылы TCP/IP хаттамасының құруымен ARPANET Интернет болып өзгертілді. Бір-бірімен қосылған түрлі алдыңғы желілер NSF-тің негізі, ал Интернет өсіп жалғасқан соң ол да Интернеттің негізі болды. 1980 және 1990 жылдардың басында, NSF АҚШ-тың көптеген жоғары оқу орындарында Интернеттің қол жетімді болуын қамтамасыз ететін бастамаға демеуші болды. Сонымен қатар, төленетін желілер (AOL, Genie, CompuServe және т.б) Интернетке қосыла бастады.

1980 жылдардың соңынан бастап 1990-жылдардың ортасына дейін бірнеше іс-шаралар орын алды, олар Интернеттің өзгеруіне әкеліп соқты. Біріншіден, оған қол жетімді кітапханалар, мектептер мен жоғары оқу орындар арқылы желі ақылы болды. Екіншіден, гипермәтіндік құжаттарға жол ашылды. Үшіншіден, АҚШ үкіметі компанияларды жеңілдету үшін Интернет қызметтерін ұсынуды жаппай пайдалану берілу туралы заңнама қабылдады. Алайда, көптеген адамдар үшін Интернетке қосылу оңай тәсіл емес.

Шамамен 1994 жылы веб-шолғыш ойлап табу әзірленді. Олардың ішінде бірінші мозаика ретінде белгілі болды. Веб-шолғыш ойы шолғытты жүктейтін және құжатты автоматты түрде көрсету болып табылады. Құжат гипермәтіндік ақпараттарды (сілтемелер) қамтитын болады. гипермәтіндік сілтемені басу кезде шолғышты команда іске қосылады, HTTP арқылы жаңа құжат ашу үшін сұрау жібереді, ол сервер арқылы жіберіледі. Құжат қайтарылып жіберіліп және шолғышта көрсетіледі. Осы уақытқа дейін пайдаланушылар хабарламаны (мейлі ол FTP, Telnet,

электрондық пошта немесе басқасы болсын) қалай жіберу керек екенін, IP-лақап ат мен мекенжайды қалай білу керек екенін түсінбеді. Бірақ веб-шолғышпен пайдаланушы тек сілтемелерді қалай басу керектігін білуі керек еді. Уақыт өте келе, веб-шолғыш интернет тілдесуінің көптеген әдістерін ауыстырды, қарапайым пайдаланушының электрондық пошта арқылы тілдесуін қоспағанда.

Біз бүгін қайда тұрамыз? Компьютерлер жүздеген миллион Интернетті құрайды. Пайдаланушылардың көбі өмір бойы үйдегі компьютерлерімен пайдаланбайды, сондықтан, нақты саны есептеп айту қиын. Егер барлық портативті құрылғыларды ескере отырсақ, онда 4 млрд. санынан асып түседі. Интернетті пайдаланушылардың нақты саны бұл алдын ала бағалау (2012 жылғы жағдай бойынша), бірақ үй компьютерлері және портативті құрылғылар арасындағы пайдаланушылар 2 млрд астам (әлемдегі халықтың шамамен үштен бірі) деп бағаланады. 12.15 суреті 1995 жылы Интернетті пайдаланушылардың өсуін көрсетеді. Ал веб ше? Интернет арқылы триллион құжаттарға рұқсаты бар. Осы құжаттардың көбі гипермәтін (HTML) құжаттары болып табылады.



Интернет пайдаланушылардың саны (млн.мен)

12.15 суретіндегі Интернет пайдаланушылардың саны 1995 жылдан 2012 жылға дейін.

Алайда көптеген құжаттардың мөлшері деректі файлдарды білдіреді (мәтіндік құжаттар, PDF құжаттар, PowerPoint құжаттар, т.б.).

Бүгін Интернет басқару негізінен Интернет желісіне қол жеткізуді қамтамасыз ету корпорациялардың қолында. 1996 жылы АҚШ-та қабылданған телекоммуникация жаңының арқасында, бұған көптеген телекоммуникациялық компаниялар кіреді. Бұл акт телефон компанияларына, кабельді компанияларға, ұялы телефон компанияларына және басқаларға интернет провайдер ретінде қызмет етуге мүмкіндік берді. Осы уақытқа дейін Интернет желісіне қатынау жұмыс берушілер мен ЖОО арқылы немесе осындай WANs CompuServe, Америка On-Line, және Prodigy төлемдері арқылы қол жетімді болды.

Осылайша, телекоммуникациялық компаниялар Интернет желісіне қатынауды бақылап отырады. Бірақ кім Интернетті өзі бақылайды? Қандай да бір мағынада, интернетті барлық пайдаланушылар қадағалап отырады, түсінікті қылып айтқанда, электрондық поштаны пайдалану, электрондық веб-беттегі деректер және біз жүктейтейтін және жіберілетін файлдар мазмұны үшін жауапкершілікті біз өзіміз жасаймыз, сондай-ақ тақталардағы біздің қатысуымыз арқылы іске асатын, Internet Control т.б. жарнамалық, веб-сайттар, блогтар да кіреді. Алайда, интернеттің кейбір аспектілері әр түрлі халықаралық консорциумдарға берілген.

Осындай бір ұйым атаулары мен нөмірлерін иемденетін Интернет корпорация атымен белгілі (ICANN). Бұл ұйым IP-мекенжайды көрсетеді және IP-дауының домен атын (бүркеншік ат) өңдейді. ICANN бөлімі Интернет нөмірлерін тағайындайтын IP-мекенжайларды бөледі. Бұл географиялық мекенжайдың орналасуына негізделген, мысалы IP-мекенжайдың бір түрі Солтүстік Америка және Еуропа мен Батыс Азиядағы бірінде орналасқан сайттарға беріледі.

Тағы бір ұйым Консорциум World Wide Web (W3C) болып табылады. Бұл ұйым веб-сайтты құру үшін стандарттар әзірлейді, оның ішіне HTML түзетуінің келесі тілдік белгілері, семантикалық желімен байланыстығын, кестелі каскадты стилі, XML-ді пайдалану ұғымдары кіреді.

Интернеттің болашағы

Интернеттің біздің қоғамға үлкен әсер ететіні бұл айдан анық. Қоғамның Интернеттен алатын артықшылықтарының тізімі өте көп. Алайда, қазіргі ағымындағы Интернетпен байланыста айқын мәселелер баршылық. Қазіргі уақытта, осы мәселелерді шешу оқытылып жатыр.

1. Коммутациялы қатынау. Көптеген адамдар телефон желісін Интерне-

тке провайдерлер арқылы кіру үшін Интернеттің қызметін пайдаланған. Телефон желісі ең бастысы айнымалы жұп сымдар жиынтығынан тұратыны мәселе болып табылады, сондай-ақ осы орта тек тілдесу трафиінің шектеулі қабілетінен қамтылады (шамамен 56,000 бит). 1 Мб файлды жүктеу жиілік жолағында 18 секундқа созылады (мысалы, шағын сурет). Сандық кабель және тікелей сандық абоненттік желілер (DSL) сияқты басқа технологиялар қазір қолда бар болғанымен, олар адамдар қоныстанған аймақта халық баяу Интернетті қолдануға мәжбүр. Бұл «технологияның соңғы милясі» деп аталды, өйткені телефон желісінің үлкен бөлігі ең талшықты-оптикалық шоғырсымды пайдаланады, бірақ бұл аймақтың арасына үйге дейін бөліну. Соңғы миляны немесе екі айнымалы сымды ауыстыру мүмкін емес. Телефонды компаниялар қазір ғана біртіндеп оны ауыстырып жатыр. Соңғы миляның технологиясы ірі қалалардағы адамдарға әсер етуі екіталай болғанына қарамастан, үлкен дәрежеде, қаланың шетінде немесе ауылды жерлерде өмір сүріп жатқан көптеген адамдар Интернетке кіру үшін телефон желілеріне сүйенеді. Бір серікті антенна арқылы қосылу, бұл мәселені шешу болып табылады. Серіктік телефон желілерінің артық өткізілу қабілеті ретінде ақпарат өте тез жүктеледі. Телефон сымын жүктеу арқылы болуы керек, бұл тек ішінара шешім болып табылады. Бұл мәселенің тағы бір келесі шешімі ұялы телефондарды пайдалану болып табылады. Алайда, ұялы телефондар желілері жиі толып тұрады, сондықтан өткізу жолағы, сайып келгенде, шектелген болып келеді. Шын мәнінде, тым көп хабарларма келген кезде 3G желілері пакеттерді түсіріп текшелейді. 4G желілері бұл мәселені шешу үшін уәде етеді, бірақ 4G тарғылмен қамтылған.

2. TCP / IP ол (32 бит 232 немесе шамамен 4 млрд. мекенжайлар комбинацияларын береді), 4 миллиард бірегей IP-мекенжайын ұсынғанын көруге болады. Алайда, барлық мекенжайлар пайдаланылмайды. Мысалы, C желілі сыныпты шағын ұйымды қарастырайық, олар 256 түрлі мекенжайларды алады, бірақ олардың барлығын қолдану міндетті емес. Тағы бір ұйым екі C желілі сыныпты талап ете алады, өйткені оларда 256 мекенжай құрылғы бар, бірақ кем дегенде 512 болуы мүмкін. NAT арқылы ұйым өздерінің құрылғыларын IP-мекенжайларын қамти алады, бірақ бұлар барлық портативті құрылғыларға жайылып кетпуі керек, оларды қазіргі уақытта Интернет қолданып жүр. Жер бетінде 7 миллиард адам барын біле отыра, біз барлық мекенжайларды тегістедік.

3. Интернет магистралі миллиардтаған пайдаланушыларды қолдау үшін құрылған жоқ, әдетте пайдаланушылар көптеген құжаттарды сұратқан кезде, олар суреттер, фильмдер және музыка, Интернет жылдамдығы баяулайды.

Кейбір мәселелерді кейбір желімен шешу үшін IPvP қолданылады (IP түрі). Алайда IPvP барлық мәселелерді шеше бермейді, сол үшін ғалымдар Internet2 технологиясын, Интернеттің жаңа пішінін зерттеп жатыр. Таяу жылдарда, бұл дереккөз қолданыла ма немесе тағы басқа құрылғыны талап ете ме екені шешіледі.

Интернет инфрақұрылым және хаттаманы жаңарту кезінде «Web 2.0» ұғымына қатысты екі қосымша ұғым қатар жүреді. Бұл Cloud есептеуі және семантикалық желі. Бұл тарауда біз қысқаша олар жайлы айтып өтеміз. Cloud есептеуі өзіне бірнеше түрлі технологияларды біріктіреді: есептеулерді топтастыру, компьютерлік желі, мәліметтерді сақтау желісі. Қызмет ретінде, бұлтта жергілікті компьютерде есептеу және сақтау мүмкіндігін қамтамасыз ету болып табылады. Сақтау бұлтты арқылы пайдаланушылар деректерге қол жеткізеді алады, себебі олар, әдетте, ол деректерге кей жағдайларда қол жеткізе алмайды, мәселен, іздеу жүйелерін қолдау үшін қайта өңделетін процессор қуатының артуы, олардың бұл жергілікті компьютерге лицензиясы жоқ, сондай-ақ үлкен көлемді файл сақтайтын сервер немесе компьютер. Негізінде, бұлт әмбебап есептеу қабілеттерін кез келген уақытта, кез келген жерде есептеулері бар арнайы қадам болып саналады. Шын мәнінде, бұлт клиент-сервер желісінің түпкісі болып келеді. Клиенттер бұлт серверіне өз бетінше кіре алуға мүмкіндіктері бар, бірақ дәстүрлі клиент сервер моделі сервер бұлттынан айырмашылығының бар екенін байқайды, бұлт пайдалануды өте үлкен ауқымда жүзеге асырады.

Бұлтқа қол жеткізу үшін әдетте, веб-портал интерфейсі арқылы жүзеге асырылады. Сондықтан, клиенттік компьютерлер өздерінің компьютерлеріне жүктелген басқа бір нақты ақпаратты қамтамасыз етуден мұқтаж болмайды. Кейбір жағдайларда клиенттер арнайы қосымшаларды жүктейді. Олар қауіпсіз есептеулерді қамтамасыз етеді (журнал талап ететін есеп жазбасы қол жеткізу үшін керек). Бұлтты есептеулердің қызметтері қарапайым сақтау мен веб-қызметтері, электрондық пошта, ойындар кең ауқымды компьютерлік өңдеу үшін жүктемені баланстайды.

Бұлтқа қолдау көрсету үшін компания аса қиын инфрақұрылым желісіне тәуелді болады. Бұл инфрақұрылым жүздеген, мыңдаған, миллиондаған пайдаланушылардың аса жоғары өткізу мүмкіндігімен жұмыс жасауы қажет. Сақтау көлемін іске қосу үшін петабайтты жақты талап етеді (келесі сақтау көлемі терабайттан жоғары). Оннан жүздеген аса кең ауқымды өндірістік процессорлар үшін сақтауда, бұлтта керек болуы мүмкін. Компаниялар инфрақұрылымды бұлттық қызметті қолжетімді ету үшін миллиардтаған долларлар жұмсайды (мысалы, Amazon, Google, Oracle). Нәтижесінде, кішігірім ақы үшін ұйым және бөлек тұлғалар мәлімет-

терді қауіпсіз сақтаумен өз бетінше өңдеулердің күшіне қолжетімділік алады (сайып келгенде, олардың мәліметтері жиі көшіріліп, қауіпсіз болуы қамтамасыз етіледі). Бұлттық есептеулер толып қалу мүмкіншілігіне ие болды, компьютер бетінде бізге өзіміздің бөлімдеріміз керек емес, бірақ оның орнына интерфейс пен басқа да сақтаулар мен есептеулердің басқа жерде орындары бар.

Бұлттық есептеулерді Интернет жасамауы керек, бірақ саудалық және тәжірибелік мақсатта, көптеген бұлттық есептеулер Интернет ретінде қолданылады. Егер компаниялар өздерінің жеке бұлттарын жасап шығаратын болса, онда оны пайдалануға шектеу қойылады, оған тек компания желісіне қосылғандар ғана қолжетімділік ала алады. Негізінде, біз мұны виртуалды жеке желімен және экстранет арқылы көре аламыз. Компания өңдеу мен сақтауды қамтамсыз етеді, бірақ VPN арқылы, компанияның ресурсты есептеулерінің пайдаланушылары оған өз бетінше қол жеткізе алады. Осылайша, олар мейлі, қайда кетіп жатсада өздерінің есептеулерін алып жүреді.

Ақыры, біз семантикалық желіні қарастырамыз. World Wide Web гиперсілтемемен қоса, өз алдына мәліметтер жиынтығын көрсетеді және ол Интернет серверінде сақталады. WWW өте кең ауқымды ақпаратты қамтиды, бәлкім, барлық адам білімін, мәселен, кітапханалар Интернет арқылы қолжетімді болуын көрсетеді. Дегенмен, ақпарат жаман негізде индекстелінеді және ұйымдаспаған.

Ол интелект үшін ақпаратты електен өткізіп және оның мағынасын түсіну үшін семантикалық Web жасанды интелект саласындағы зерттеушілер бастаған үрдіс болып табылады, ол білімді ұйымдастыру және автоматтандырылған іздеу бөліктерін ақпарат арқылы өткізу ғана емес, пайдалы бөліктерді таба отырып пайдаланушы адамға көрсету.

Мәселен, мысал ретінде алсақ сіз Гавайи аралында дем алғыңыз келеді. Әртүрлі веб-сайтты редакциялау мен іздеу жүйесі арқылы сіз сапарды, қонақ үйді брондап, көлікті жалға алып, өзіңізге Гавайи аралын аралап білуге орын мен қызықты оқиғаларды аша аласыз. Бұл үрдісті қаншалықты автоматтандыруға болады? Ойлап көрейік сізде күні бұрын алынған бот бар (ақпаратты қамтамасыздандырудың сөзге берілген бөлігі). Сіз өзіңіздің қызығушылықтарыңызды атап кеттіңіз, нені ұнатасыз, ал нені ұнатпайсыз. Сіз өзіңіздің репрезентті-бот күнтізбеңіздегі маңызды күндерді көрсетесіз, ол күндер сіздің сапарыңыздың күндеріне шектеу қояды. Бағалық шектеу мен қоса басқада критерийлерді көрсете аласыз. Күні бұрын алынған бот осы барлық ақпараттарды пайдалана отырып, әртүрлі веб-сайттармен тілдеседі, сіз үшін барлық демалысты брондауға, жергілікті жерде сізге уақыт өткізіп баратын жерлеріңізді

көре алатын карта берілгенге дейінгі мәселелерді қамтамасыз етеді. Сонымен қатар, бот уақыт жақындығын, оларды үйлестіруге, оқиғаларға айтуға болатын түрлі орындар арасындағы туристік, күн/түн уақыты және басқа да факторларын ескереді.

Семантикалық веб-сайт бірнеше әртүрлі компоненттерден тұрады. Веб-сайт мәліметтері қазіргі уақытта онтология саласына ұйымдастырылады: білім негізін қамтитын ақпаратты ғана емес және сол ақпаратпен қалай пайдалана алу мен түсіндіре алу жасақталған. Кішігірім ақпараттар, интеллектуалды агенттер әр түрлі міндеттірді өңдеуден құрылған, мысалы жоспарлану сияқты, сауда жөніне қатысты шешімдерді қабылдау. Жаңа түрлі тілді сұраулар агент пен онтология арасындағы қарым-қатынасты жеңілдету үшін құрылады. Сонымен, біз осы жаңа технологиялар World Wide Web семантикалық төсемінің жақын жылжуы, онлайн ресурстарға оңай қолжетімді, ал адам қатысуынсыз қайсысының білімін шамадан тыс пайдалануға болады деп үміттенеміз. Семантикалық желі 2001 жылдан бастау алады, қазіргі уақытта ғана мұндай ойлар барлау жиынтығының мәліметтері және медициналық зерттеу сияқты салаларда пайдаланылады. Оларды тәжірибе жүзінде күн сайын қолдану үшін бірнеше онжылдықтар не одан да көп жылдар өтуі мүмкін.

13. Бағдарламалық жасақтама

Бұл бөлім бағдарламалық жіктеу мен технологиялар кіріспесінен басталады. Бұл бөлімде ең бастысы бағдарламалық жасақтамаларды басқаруға, сондай-ақ Windows бағдарламалық жасақтамасында Linux-ті орнатуға көңіл бөлінеді. Linux-тегі талқылау басқару бағдарламалық жасақтама пакеттерін пайдаланумен байланыстырылады және ашық бастапқы бағдарламалық құралын орнатады. Тарау кіріспе деңгейде Apache веб-сервердің уақытылы орнатылуына және конфигурациясына ерекше назар аудартады, сервер бағдарламалық жасақтаманы ескеріп отырып жасалады.

Тарауды жасақтаудың мақсаттары

- Бастапқы және еркін бағдарламалық жасақтама және бағдарламалық жасақтаманы жіктеу түрлері талқыланды.
- Шеберлік орнатудан және пакет менеджерінен алынған бағдарламалық жасақтаманы орнату процесі сипатталады.
- Бағдарламалық жасақтаманың ашық бастапқы коды бар Linux-те конфигурация арқылы орнатуды нақты құралдармен түсіндіру
- Танымал сервердің бағдарламалық функциясын сипаттаңыз.
- Apache веб-серверін орнату және оның конфигурациясымен танысу.

Мұнда біз бағдарламалық жасақтаманың түрлеріне қарап АТ маманын білу қажет екенін қарастырамыз. Бағдарламалық жасақтама тұжырымдамалары мен терминология жүзеге асырылады, сонымен қатар біз бағдарламалық жасақтаманы техникалық қызмет көрсету және орнату бойынша бағытталатын көрсетеміз. Біз осы тарауды сервер параметрлерін орнату және техникалық қызмет көрсетуді тексерумен аяқтаймыз.

Бағдарламалық жасақтамалардың түрлері

Бағдарламалық жасақтама термині біз компьютерде жүзеге асырған

бағдарламаларға жатады. Біз бағдармалық жасақтама терминін физикалық, материалдық пішіндегі бағдарламада жоқ болған соң қолданамыз. Жадта сақталу барысында олар электрлі тоқ ретінде, ал дискте сақталғанда магнитті заряд ретінде жүзеге асады. Сондықтан, біз аппараттық жасақтамадан бағдарламалық жасақтаманы айыра аламыз, оларды біз көрсетіп, таңдап, ұстай аламыз.

Бұл айырмашылық айқын болып көрінуі мүмкін. Дегенмен, шын мәнінде, бұл ауызға айтуға тым жеңіл. Біздің компьютерлер ортақ мағыналы, ал басқа кез келген бағдарламаны іске асыра алады дегенді білдіреді. Ең алғашқы 1930 жыл мен 1940 жыл аралығындағы компьютер ортақ қолданыста болмаған, керісінше, белгілі бір интегралды математикалық есептеу анализін немесе кодтың бұзылу үрдісін анықтау үшін арналған. Бүгінде, біз өзіміздің кейбір құрылғыларымыздың жады мен процессоры бір мағынада екенін көре аламыз (мысалы, навигациялық жүйелер, mp-3 плеерлері, ойын платформалары). Сонымен қатар, бұл ерекшеліктерге қарамастан компьютерлер кез келген бағдарламаны осы платформа негізінде іске қоса алады. Ал басқа қырынан алып қарағанда, бағдарламалық жасақтама сызба түрін қолдану арқылы да жүзеге асырылады. Еске сала кетейік, ең төменгі деңгейде компьютер AND, OR, NOT, XOR жинақтамалары ретінде болады (мәліметтерді ауыстырып тұру үшін сымдармен қоса). Сонымен, біз басқа компьютерлер бөліктерін жағалай өтіп, осы бағдарламаны сызба түрінде жүзеге асыруымыз мүмкін. Бағдарламаны аппараттық құралдарында сөзсіз қамтуды, жиі қосалқы орналасатын бағдарламалық жасақтама ретінде айтуға болады (немесе желілік іске асу). Сол сияқты, кез келген қондырылған бағдарлама, компьютердің жалпы максатында және сызба түрінде бағдарламалық тілмен жазылады.

Бұл концепсия аппараттық және бағдарламалық балама ретінде танымал. Сондай-ақ, бір бағдарламалық жасақтаманың көмегімен шақырылған басқа құрылғыны, біз айыра алатындай етіліп жүзеге асарады, бірақ жүзеге асырылуы нақты солай екенінде емес. Біз өзіміздің бағдарламаларды бағдарламалық жасақтама ретінде қамтыдық, ол аппараттық құрылғы ретінде қамтылмайды, өйткені бағдарламалық жасақтама икемді болып келеді. Біз оны кейінгі уақытта өзгерте аламыз. Алайда, баламалы бағдарламалық жасақтамаға қарағанда, қосалқы бағдарламалық жасақтаманы құру көп қаржыны талап етеді. Басқа қырынан алғанда, CPU процесі іріктеуді істейді, ол бағдарламаны жасауға уақыт қосады. Бағдарлама тезірек бір жазу бағдарламасына қарағанда, жылдам жақсы мәміледе іске қосылып бағдарламада жүзеге асырылады. Соған қарағанда, кез келген бағдарлама теориялық тұрғыда бағдарламалық

жасақтама немесе қосалқы бағдарламалық жасақтама ретінде іске асады. Өнім өзіндік құнының арзанырақ болуы немесе жылдамдық талпынысына икемді болуына ауқымды түрде керектігін арттыратын шешімге әкелініп соғады. Бүгінде көптеген бағдарламалар бағдарламалық жасақтама түрлері арқылы жүзеге асады. Манағы қамтылған шілтелер басқа құрылғылардан табылған бағдарламалар ішіне кіреді, ол дегеніміз, бұғатталуға қарсы жүйесі мен көліктегі жүктеме компьютерінің бүрку жүйесі болып табылады.

Біз өзіміздің бағдарламаларымызды бағдарламалық жасақтама ретінде сақтағанда жөн санаймыз, өйткені бағдарламалық жасақтаманы өзгертуге болады. Бүгінде көптеген бағдарлама өнімдері шығарылған, содан кейін олар қайта-қайта қарастырылған. Біз бағдарламалық жасақтама үшін шығарылым бейнесін көрсететін тізбектеме дамытып шығардық. Көптеген бағдарламалық жасақтамалардың атаулары тараудан кейін көрсетілген екі телефон нөмірінің жиынтығынан тұрады. Бірінші нұсқада шығарылым негізгі болып, ал екіншісі кәметке толмаған түрде болып бөлінеді. Мысалы, Mozilla Firefox 5.2 түрі бесінші болып келеді, Firefox ең негізгі түрі, ал екіншісі кішігірім 5 түрге жатады. Болмашы шығарылым әдетте қателер түзетілгенін жырық қауіпсіздігі қосылғандығын білдіреді. Негізгі релиз әдетте бағдарламалық жасақтаманың өте көп функциялардан тұратынын білдіреді, сонымен қоса, пайдаланушының жаңа графикалық интерфейсі (GUI). Негізгі шығарылым жаңа кодпен көшіртілуі мүмкін. Ұсақ релиздер айына бір немесе екі рет, ал негізгі іріктемелер бөлігі жылына бір немесе екі рет шығарылуы мүмкін. Болмашы іріктемелер тағы бір жүзеге асу пакеттердің қайта жаңаруы болып табылады. Дегенмен, пакеттердің қайта жаңаруы операциялық жүйе жаңаруының жиынтығын жүзеге асыруда пайдаланылады. Жалпы, біз кез келген жүйені бағдарламалық жасақтама немесе қолданбалы бағдарламалық жасақтама бағдарламалық құралы арқылы санаттаймыз. Қолданбалы бағдарламалық жасақтама кейбір тапсырманы орындау үшін жұмыс істейтін қандай бір жұмыс бағдарламаларынан тұрады. Бағдарламалық жасақтаманың қолданбалы түрі бөлшектелінген мансапта болады, өйткені әрбір мансап бағдарламалық жасақтама оны өзі қолдайды. Бағдарламалық жасақтама өндірілімділігі бәріне пайдалы болатын тиімді қосымшадан тұрады. Бағдарламалық жасақтама өндірілімділігіне мәтіндік процессор, бағдарламалық жасақтамалардың презентациялары, электронды кестелер, дәрекетер базасын басқару жүйесі, күнтізбе бағдарламалары, мекенжай кітабы және органайзер кіреді. Сурет салу бағдарламасы осы топтамаға кіре алады, бірақ сызу көркем және графикалық дизайн топтамасына жатады. Басқа қолдан-

балы бағдарламалық жасақтама Интернетті пайдалану негізінде болады. Бағдарламалық жасақтаманың бұл сыныбында клиент өзіне электрондық поштаны, веб-шолғышты, FTP-клиент құраламарын кіргізеді. Сонымен қоса, бағдарламалық жасақтамада компьютерлік және көңіл көтеретін ойындар бар (мысалы, видео-плеерлер, DVD-плеерлер компакт-дисктегі күйтабақ ойнатқыштар); бірақ бағдарламалық жасақтаманың өндірілімдігі оларды қоспайды, олар барлығы үшін қолданылады. Бағдарламалық жасақтама өндірілімдігі көлденеңді бағдарламалық жасақтама болып аталады, бұл бағдарламалық жасақтама ұйым бөлімдерінің барлығында қолданылады. Тік бағдарлама жасақтамасы бөлек топтар ұйымымен қолданыста болады. Мысалы, музыканттар қолданылатын музыка, секвонсорлар, цифрлі жазылатын құрылғылар бағдарламалық жасақтама сыныбы болады. Бағдарламашылар бағдарламалау тілін компиляторда ғана емес даму платформасында да пайдаланады, сондай-ақ көмек коды, кітапханалар коды, өзгерту бақылау коды және визуализация коды түрінде қолданылады. Кинематографистер мен суретшілер, фото редакциялауда және манипуляциялар редакциялауда видеоны пайдаланады, сондай-ақ дыбыс редакциялау үшін бағдарламалық жасақтама керек. Сандық тізілімдер үшін мамандандырылған бағдарламалық жасақтама қолжетімді болып бухгалтерлік бағдарламалық жасақтамада алуан түрі болады.

Жүйелік бағдарламалық жасақтама операциялық жүйенің бөлігі болып табылатын бағдарламалардан тұрады. Жүйелік бағдарламалық жасақтама тікелей компьютер жүйесін өзі қолдайтын бағдарламалық жасақтама болып табылады. Мұндай бағдарламалық жасақтама көбінесе пайдаланушы сұранысына қарағанда, автоматты түрде іске қосылады. Жұмыс істейтін операциялық жүйеде кейбір фондық данада тәртіп бар. Олар жиі демондар немесе қызмет деп аталады. Олар шешім қабылдаудан бұрын орын алатын оқиғаларды күтіп тұрады. Операциялық жүйенің өзге бөліктерінде жоспарлаушы негізінде орындалады. Мысалы, сіздің вирусқа қарсы бағдарламалық жасақтамаңыз 24 сағат сайын іске асады және кез келген жаңартулар үшін веб-торапты тексереді және аптасына бір рет жүргізілуі мүмкін бағдарлама болып табылады. Алайда, көбінесе коммуналдық деп аталатын басқа да нысандар, талап бойынша орындалады. Мысалы, пайдаланушы қатты диск өнімділігін арттыру үшін Дисктің дефрагментациялау құралын іске қосуына болады. Жоғарыда көрсетілгендей, бұл жоспарлаушы негізінде жүзеге асырылуы мүмкін, бірақ вирусқа қарсы бағдарламалық жасақтамасының басқа коммуналдық бағдарламасы болып табылады.

Операциялық жүйенің әр түрлі деңгейі, сондай-ақ, олардың пайдала-

нушымен өзара аппараттық, бағдарламалық жасақтамасы 13.1 суретте көрсетілген. Фигура ламинатталғандығын көрсетеді, сондай-ақ пайдаланушының жоғарғы жағында жұмыс істейтін компьютерлік жүйесінде ең төмен деңгейде орын алады. Пайдаланушының әрбір әрекеті төмен қабатты жылжыту кезінде қарабайыр операциялармен ыдырайды.

Операциялық жүйенің негізгі компоненттерін ядро деп атайды. Компьютер бірінші жүктелген кезде ядро жүктеледі. Өзегінде процесті басқару, ресурстарды басқару және жадты басқару үшін жауапты компоненттері болады. Онсыз сіз жаңа бағдарламаны іске қосу үшін немесе бағдарламаны тиімді орындай алмайсыз. Ядро аппараттың жоғарғы жағында орналасқан, аппараттық және бағдарламалық жасақтама арасындағы интерфейс болып табылады.

Операциялық жүйенің тағы бір бөлігі құрылғы драйверлер жинағы болып табылады. Құрылғы драйверінің бағдарламалық жасақтамасы аса маңызды бөлік болып табылады. Пайдаланушы бағдарламалық жасақтама мен қосымшаларын жалпы аппараттық құрылғылардан көруге болады, мысалы енгізу құрылғысы сияқты, үлес қосқан күйін тексеру, қайта іске қосу командасы ретінде жауап береді. Алайда, әрбір құрылғы бірегей пәрмендерді талап етеді. Осылайша, операциялық жүйе нақты командаларға неғұрлым жалпы пәрмендерді алады. Аударма бағдарламалық жасақтамасының құрылғы драйверлері арқылы жүзеге асырылады. Бірнеше құрылғылар драйверлері микробағдарлама, бағдарламалық жасақтаманың жүргізуші құрылғылары сияқты BIOS ROM шырмауында да болады. Жаңа аппараттық құрылғы сатып алынған кезде, басқалары, Интернет немесе CD-ROM-ға түсірілу тиіс, ал көптеген танымал құрылғы драйверлері операциялық жүйесінде алдын ала жүктеледі. Жаңа аппараттық құрылғы пайдаланылуы мүмкін болғанда, кез келген жағдайда, құрылғы драйверіне орнатылуы тиіс. Онда жүздеген жүргізушілерге мыңдаған (1,5 млн астам, driverguide.com сәйкес) қолжетімді сөзбе табылады және көпшілігі пайдаланатын болады, ол кезде олардың барлық дискілерін пайдаланушымен толтырып қажет емес. Жүйелік бағдарламалық жасақтамасының басқа компоненттері (қайтадан 13.1 суретін қараңыз) ядроның «жоғарысында» тұрады.

Ядроның үстіндегі компоненттер пайдаланушының қоршаған ортаға бір бейімделігін (Shell немесе пайдаланушының жұмыс үстелі), жүйелік бағдарламалар мен қызметтер демонын қамтиды. 9 тарауда айтылғандай, Linux Shell пайдаланушылар өз лақап атты және айнымалылар жасауға мүмкіндік береді. Desktop элементтер тіркесімдерін, тақырыптарды, мәзірлерді және фонды, Бастау мәзірінде, және Windows 7 жұмыс үстелі гаджеттерінің құралдар тақталары мен бағдарламаларының пай-

да болуын қамтиды. Амалдық жүйе компоненттері (жүйелі бағдарламалық жасақтама).

Пайдаланушы жүйесінің жұмысын бақылауға және жақсартуға мүмкіндік беретін коммуналдық бағдарламалар жүйесі. Біз процестер туралы әңгімелеген кезде олардың кейбіреулері талқыланып, жүйе өнімділігін жақсарту үшін құралдар сіздің қатты дискіңізде дефрагментациялау және вирустар мен басқа да зиянды файлдарды сканерлеу жасалынады. Олар (көбінесе олар сатып алушылардың талабын және де көптеген пайдаланушылар олардың қатысуын ескермеуі мүмкін) операциялық жүйесімен бара алмайды, өйткені әдетте, жүйе коммуналдығын жеке орнату керек.

Қызметтер немесе демондар операциялық жүйеде жүктелген және талданған, бірақ фонда жұмыс істеп тұрғанда, әдетте бастау жүйесі бағдарламаларды іске қосады. Бұл бағдарлама әлі де белсенді кезде, ол процессордың уақыт кетуі мүмкін емес екенін білдіреді, бірақ соның өзінде шақырылмайды. Ағымдағы операциялық жүйеде нақты іс-шара болған кезде, іс-шараны өңдеуге қажетті қызмет немесе демон қоңырауы болып табылады. Кез келген операциялық жүйеде көптеген қызметтер бар. Біз 11 тарауда Қызметтер Параметрлер қызметтерде қамтылғанын және жүктеу кезінде қызметтер жасағанын көреміз.

Оған қарамастан, жүйелі бағдарламалық жасақтама және қолданбалы бағдарламалық жасақтамалы сервер бар. Серверлер бірнеше атауға, Веб серверлерге, электрондық пошта серверлеріне және дерекқор серверлеріне кіреді. Біз Apache веб-серверіне баса назар аудара отырып, серверлер және сервер қызметтерін қарастырамыз.

БАҒДАРЛАМАЛЫҚ - Ілеспелі терминология

Бұл бөлімде біз қысқаша қосымша шарттарды қарастырайық, олар бізге бағдарламалық қамтамасыз етуді түсінуге көмектеседі.

Платформа үйлесімділікті сипаттағанда, бағдарламалық жасақтаманың бір бөлігі орындалуы мүмкін. Біздің компьютерлер басқару түрінде жұмыс жасағандықтан, көптеген бағдарламалық өнімдер бағдарламалық жасақтамалардың аталған компьютер платформасына арнайы жинақталуы тиіс. Бұл жерде үш сұрақ қойылған. Біріншіден, бағдарламалық жасақтама аталған компьютердің процесс сыныбы үшін жинақталуы тиіс. Pentium және Intel процессоры, мысалы MIPS процессорына қарағанда басқарудың түрлі жиынтығына ие. Егер кімде-кім Pentium процессорына арнап бағдарлама жазып әрі оны жинаса, жинақталған бағдарлама MIPS процессорында жұмыс істемейді.

Құрылғы мен операциялық жүйе арасындағы ұқсас сипаттамаларға байланысты аталған операциялық жүйе платформасыда жұмыс жасауы тиіс. Linux-тің басқаруымен жүретін Pentium процессоры бар компьютер Windows-қа арнап жинақталған бағдарламаны тап осындай процессорды қолдана отырып орындай алмайды. Сондықтан бағдарламалық жасақтама процессор үшін де және операциялық жүйе үшін де жинақталуы тиіс. Және, ақыр соңында, бағдарламаларды іске қосу үшін қосымша талаптар пайда болады немесе тиімді жұмыс істеуіне, мысалы жедел жадтың аз мөлшердегі саны немесе қатты дисктегі орын көлемі не бейне карталардың айқындалған түрлері.

Қайырмалы қосылғыш дегеніміз бағдарламалық жасақтаманың ескі нұсқасынан ақпаратты файлдарды қолдану және оқу болып табылады. Бағдарламалық жасақтамалардың көптеген сатушылары қайырмалы қосылғышты қолданғандықтан, тұтынушы ескі мәліметтерді жоғалтып аламын деп аландамай бағдарламалық жасақтаманы жаңартуға болады. Тұтынушы қайырмалы қосылғышты сақтамай, жаңартудан мүлдем бас тартуына болады. Қайырмалы қосылғышты қолдау үлкен мәселе тудыруы мүмкін, өйткені көптеген қызметтер ескі нұсқаулардың негізінде жарамсыз болып қалады әрі жаңа қызметтерге кедергі келтіруі мүмкін, сондықтан оларды сақтамауға болады. Қайырмалы қосылғыш ескі бағдарламалық жасақтама жұмыс істейтін жаңа процессорға да енгізіледі. Ескертеміз, бағдарламалық жасақтама нақты бір процессорға арналып жинақталады. Егер процессордың жаңа нұсқасы босатылса, онда ол қайырмалы қосылғышты қолдайды ма? Қолдамаған жағдайда ескі компьютері бар пайдаланушының компьютеріне жаңа процессор үшін жаңа бағдарламалық жасақтама қою керек. Intel 8086 бастап Pentium ең соңғы үлгісі арқылы өнімнің сызбасымен x86 арасында қайырмалы қосылғышты сақтап қалды. AppleMacintosh өзінің процессорларымен қайырмалы қосылғышты қолдамайды.

Компания бағдарламалық жасақтамалардың жаңа үлкен және кіші өнімдерін шығарғанда, жаңартулар жүргізіледі. Бағдарлама жаңа өнім үшін компанияның веб-сайтын ашқан кезде жаңартулар автоматты түрде жүзеге асырылады. Егер ол табылса, пайдаланушының тіркелуін күтпей-ақ жүктеліп, орналасады. Бұл қызметті кез келген бағдарламалық жасақтамада орналастыруға болады, яғни бағдарламалық жасақтама сізден жаңарту үшін рұқсат сұрайды әрі егер де жаңарту керек емес жағдайда оны тоқтатуыңызға болады. Дегенмен жаңашаландыру қауіпсіздіктің тесіктері мен қателіктерін түзетеді. Компания сізге жаңартуды ұсынған

кезде оны қолдану не қолданбау өз еркіңізде.

Патчи коды бағдарламалық жасақтамасының бір бөлігінің кейбір мәселелерін түзетуге көмектеседі. Ережеге сай оларда жаңартулар қарастырылмаған, бағдарламалық жасақтама нұсқасының нөмері және бөлек тұруына байланысты сандармен саналуы мүмкін. Жабыстырғыш уақыттың кез келген сәтінде шығарылуы және бағдарламалық жасақтаманың тұрақтылығына байланысты жиі не сирек босатылуы мүмкін.

Бета-релиз (немесе бета-тест) релиздің келесі шығарылымына дейін қолданылатын, бағдарламалық жасақтаманың жаңа нұсқасында қолданылатын өрнек болып табылады. Бета-релиз тізімге бағдарламалық жасақтаманың ерекшеліктері туралы кері байланысты қамтамасыз ететін әрі бағдарламалық жасақтаманы тексеруге және қателерін табуға көмектесетін сенімді деген пайдаланушыларын жібереді. Компания кері байланысты қолдана отырып қателерді зерттейді және бағдарламалық жасақтаманың келесі нұсқаулығын жақсарту мақсатында қолданады. Бета-релиз бірнеше айдан кейін немесе толық босалудың алдын-ала уақытынан бір жыл бұрын пайда болуы мүмкін.

Орнату жаңа бағдарламалық жасақтаманы алу және оны сіздің компьютеріңізге орналастыру болып табылады. Орнату процесі қатқыл дискке жай ғана кодты көшіру болып келмейді. Ережеге сай орнату файлдарда қателерді табу үшін тестілеу, түрлі тізімдемелерге файлдарды қойғанда әрі операциялық жүйедегі жалпы файлдарды тестіледі талап етеді. Бүгінгі таңда орнату ARD сикырлы орналастыруды қолдана отырып жеңіл түрде қойылады. Дегенмен, кейбір бағдарламалық жасақтаманы орналастыру оңайға соқпайды. Бұл сұрақ бағдарламалық жасақтама басқармасында қарастырылады. Пайдаланушы тығыздалған CD-ROM * немесе Интернетті сөндіру арқылы бағдарламалық жасақтаманы орнатады. Кейінгі кезде орналастыруды жүктеме деп те атайды. Жүктеу термині Интернеттің кейбір серверлерінен файлдарды көшіру әрекетіне жатады, мысалы әнді жүктеу. Интернеттен жүктеу кең көлемде тараған, өйткені көптеген адамдарда шапшан түрдегі Интернет желісі бар.

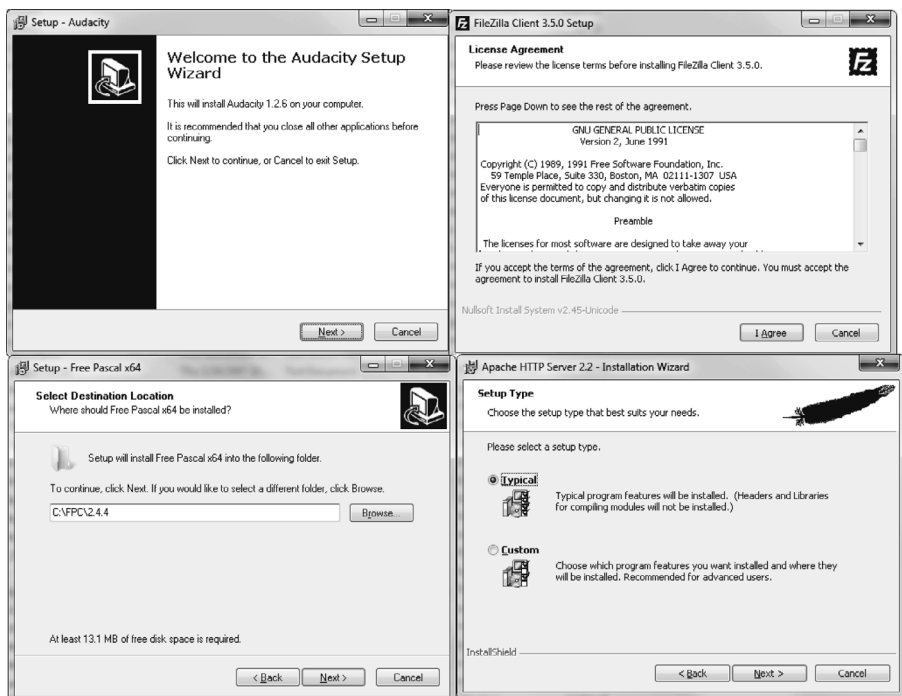
Бағдарламалық жасақтаманы сипаттаудың бірнеше категориясы бар. Сатушыдан жекеленген не коммерциялық бағдарламалық жасақтаманы сатып алуға болады. Осындай бағдарламалық жасақтаманы коммерциялық түрде сататын компаниядан (бағдарламалық жасақтама үйі деп аталады) немесе үйде өзінің дербес бағдарламалық жасақтамасын өндіретін компаниядан не бағдарламалық жасақтаманы өндіру үшін жалдаған кеңесшіден (не компания кеңесшісі) сатып алуға болады. Бағдарламалық жасақтаманы сатып алғанда, ол сізге екі не үш затты ұсынады. Біріншіден, сіз бағдарламалық жасақтаманың өзін орындалған код

түрінде аласыз. Егер сіз бағдарламалық жасақтаманы Интернет арқылы не дүкеннен сатып алсаңыз, онда бағдарламалық жасақтама орнату бағдарламасы арқылы орнатылады. Ал егер сіз оны кеңесшіден сатып алсаңыз, онда олар орналастыруды өзіңше орындайды. Екіншіден сіз бағдарламалық жасақтаманы қолдану үшін лицензия аласыз. Үшіншіден сізге осы сатып алулар пайдалы ресурстарға жол ашады, яғни техникалық қолдау, интерактивті анықтамалар және қандай да бір қосындылар.

Бағдарламалық жасақтаманың басқа да санаттары еркін бағдарламалық жасақтаманы суреттейді, яғни сіз оны тегін түрде ала аласыз. Еркін деген бағдарламалық жасақтамаларды белгісіз уақытқа дейін және қалай болсын солай қолдану емес, сіз оны төлем жасамай-ақ қазіргі сәттен бастап қолдана алсыз деген мағынаны білдіреді. Бағдарламалық жасақтаманы ең бір кеңінен тараған түрі Интернет желісінде тегін-қызмет санаты деп аталады. Shareware бағдарламалық жасақтамалардың сынамалы нұсқасын береді. Компания тегін қызметті пайдаланушылар арасында бағдарламалық жасақтамаларды сатып алудың бір амалы ретінде қолданады. Сынамалы нұсқаулықта бірнеше шектеулер қойылады, яғни қолданыстың берілген саны, берілген күндер саны не ерекшеліктерге шектеу қойылады. Егер сіз тегін қызмет арқылы бағдарламалық жасақтамаларды орнатсаңыз, онда тіркелу лицензиясын сатып алып оның жиі толық нұсқасына дейін жаңартып отыруыңызға болады. Бұл ережеге сай бағдарламалық жасақтамалардың «бұғатты ашу» қызметінде қолданылады. Сіз тегін қызмет көрсететін нұсқаулықты жүктейсіз. Сіз оны қолданасыз. Сайтқа кіріп, оны сатып алып (несиелік карталар арқылы), лицензияға ие боласыз. Осы лицензияны енгізесіз енді міне бағдарлама тегін қызмет көрсетуден коммерциялық бағдарламалық жасақтамаға ауысады.

Екінші жағынан, кейбір бағдарламалық жасақтамалар шын мәнінде тегін. Олар бірнеше санатқа бөлінеді. Біріншіден, тегін бағдарламалық жасақтамалар. Осындай бағдарламалық жасақтамалар сатып алу санатына кірмейді. Freeware ескіріп қалған бағдарламалық жасақтама, оны жаңадан құрылған бағдарламалық жасақтама ауыстырды не ол өндірушісіне (жеке тұлға не компания болсын) керек емес бағдарламалық жасақтама. Оны орнатқаннан кейін сіз оның бірнеше келісімдерімен байланыста боласыз. Бұл қоғамдық мүлікке ауыстырылған бағдарлама. Еркін қолданыстағы барлық заттарды қолдануға болады, алайда сіз қандай да бір құқықты пайдаланып таластын жоқтығын жақсы сезінесіз.

* оптикалық дисктерге дейін орнату жиі дискеттен орындалады. Дегенмен дискетте тек 0 ғана сақталуы мүмкін.



Windows орнатудағы түрлі терезелер

Пайдаланушы орнату терезелері арқылы орналастырудың қарапайымдылығымен қатар бірнеше заттарды білуі тиіс. Ең маңыздысы бағдарламалық жасақтаманы қосу үшін қажетті ресурстардың бар болуы. Ең маңызды ресурстар оперативті жадтың жеткілікті болуы әрі шапшаң процессор және қатқыл дисктергі бос орындардың жеткілікті болуы болып табылады. Бағдарламалық жасақтаманы орнатпас бұрын сіз орналастырудың талаптарын тексеруге болады. Және де бағдарламалық жасақтаманың дұрыс нұсқасын орнатып жатқанызды ескеру кересіз. Бағдарламалық жасақтаманы орналастыру құқығына ие екеніңізге көз жеткізуіңіз керек. Сіз әкімшілікке жатасыз ба? Сіз жүйеге әкімшілікке орнатуды орындауға кірдіңіз бе? Немесе пайдаланушы ретінде ме?

Linux-те бағдарламалық жасақтаманы орналастырудың бірнеше түрлері қарастырылған. Біріншіден, Windows-та көрсетілетіндей, жүктелген бағдарламалық жасақтаманы орналастырудың шеберлігін қолданады. Екінші айналым Windows сияқты CD-ROM-да (не DVD) бағдар-

ламалық жасақтаманы орналастыру. Компакт-дискті орналастыру үшін орналастыратын компакт-диск керек және Орнату бөлшектерін қолдану әрі бағдарламаны жою тиіс (Applications мәзірінде орналасқан).

Linux-тің басқа нұсқасы бұйрық жолдарында бағдарламалық пакеттерді орналастыруы болып табылады. Қолдануға болатын бірнеше амалдар да бар. Бұйрық жолдарының ең қарапайымы бағдарлама пакеттеріне менеджер арқылы ену амалы. RedHat-та екі жолы көрсетілген, бірнеше (Updater сары ит, Modified) және минут ішіндегі айналым (RedHatPackage менеджери). RPM бағдарламасы файлдар айналымы арқылы жұмыс істейді. RPM файлдары мынадай пакеттерден тұрады: атқарушы бағдарламалардан және олармен байланысты файлдардан тұратын сығылысқан файлдар (ы) мұрағаты және файлдарды қосуға қажетті сипаттамалар кіреді. RPM бағдарламасы тәуелділікті тестілеуден өткізіп, кез келген файлдың жоқтығы туралы пайдаланушыға ескертіп отырады. Бағдарлама минутына бірнеше айналымдарды қолданады, бірақ сіздер үшін тәуелділікті өндіреді.

Минут ішіндегі айналымның көмегімен бағдарламалық жасақтаманы орналастыру үшін `-ipackagename.rpm` мин. нұсқаулық қолданылады. Ақпаратталмаған бағдарламалық жасақтама үшін минут ішіндегі айналым сияқты `-e` (өшіру үшін) қолданылсын, яғни `-ePackageName`. Пакетті жаңарту үшін `-u` басыңыз. Сонымен қатар сіз белгіленген пакеттерде немесе әлі де белгіленетін пакеттерде (RPM файлдары ашық тұрғанға дейін) минут ішіндегі айналым арқылы көптеген ақпараттарды табу да мәліметтердің базасына сұраныс жіберу үшін `-q` қолданасыз. Сіз `-q` және `-l` қосу арқылы аталған пакеттегі файлдардың тізімін немесе `-q -a` және опцияларды қосу арқылы нақты белгіленген пакеттер тізімін ала аласыз. Минут ішіндегі айналым қоймалары интернет желісінің көп жерлерінде қолжетімді болып келеді, яғни олар `HTTP://centos.karan.org/EL4/қосымша/тұрақты/i386/RPMS/`. Бұйрық жолдарындағы RPM ғана қолданбай, сіз интерфейсті графикалық менеджер пакеттерін де қоса аласыз.

Yum Linux пайдаланушылары үшін Дьюк Университетінің өнімі болып табылады. Yum орналастыруды орындау үшін минут ішіндегі айналымдарды шақырады; дегенмен ол қажеттілік туындаған кезде кез келген тәуелді пакетті орнатуға және тәуелділікті іздестіру мүмкіндігіне ие болады. Минут ішіндегі айналым сияқты ням бағдарламалық жасақтаманы орналастырып жаңарта алады, сонын ішінде Linux ядросында. Бұл оны минут ішіндегі айналымның көмегімен әр қайсысын қолмен жаңартудың қажеттілігін тудырмайтын бір топ машиналарға жеңіл түр-

де қолдауына болады. Yum қолдану өте оңай; ол жұмыстың біраз бөлігін өз бетінше орындайды (пакеттерді жаңарту және жаңасын орнату үшін қыруар уақытты жоғалтуы мүмкін). Қарапайым ням бұйрығы тақырыпты белгілейді. Бұл жерде пакет атауы бағдарламалық жасақтама болып келеді. Басқа да ням бұйрықтары (орнату орынына) өздерін тізімге ("Foobar" ням тізіміндегі сияқты белгіленген пакеттің атауын алады) chkconfig ням (Yum әр түн сайын жаңартуды орындауды жоспарлайды), атауды жою нямын (барлық тәуелділікті алып тастап бүтін пакетті жою үшін), -у тақырыпты белгілеу нямын (кез келген сұраққа иә/жоқ нұсқасынан «иә» деген жауап алынбайды, орналастыруды жасайды, осы орналастыру адамның көмегісіз жасалады).

Linux-те орналастырудың ең бір қиын түрі бастапқы кодтан бағдарламалық жасақтаманы белгілеу болып табылады. Бағдарламаның бастапқы коды жоғарғы деңгей тілімен жазылған код болып келеді (мысалы, C++ немесе Java). Бұл бағдарлама тікелей компьютерде орындалмайтындықтан, ол бірінші болып жиналады (біз жинақтаманы келесі тарауда қарастырамыз). Linux-тағы көптеген қолжетемді бағдарламалық жасақтамалар ашық бастапқы кодтан тұрады. Бастапқы кодты бағдарламалық жасақтаманы орналастыру үшін сіз бағдарламалық жасақтаманың атқарушылық форматта қолжетімдігіне не орналастыру шеберлігінің көмегіне немесе минут ішіндегі пакеттер ретінде байқай аласыз.

Бастапқы кодты орналастыру бағдарламалық жасақтама жинақтары қолмен ашуды талап етеді. Ашық бастапқы кодты бағдарламалық жасақтамалардың көп бөлігі мұрағатқа жіберіледі. Файлдарды мұрағаттаудың бірнеше механизмдері бар, Unix арқылы кең тараған түрі мұрағат негізінде бағдарлама белгісінде қарамай атауы белгіленген. Қарамай бағдарламасы басында таспада сақталатын файлдарды мұрағаттауға арналған еді. Дегенмен бұл әдіс оңай әрі қарапайым сондықтан көптеген программистер осыны қолданады.

Қарамай файлы .tar-ды кеңейту ретінде қолданады. Сонымен қатар бетондау бірге іске асырылғаннан кейін қабаттар орындарды үнемдеу үшін қыстырылады және интернет арқылы қабаттарды түзу барысын тездетуге көмектеседі. Бүгінде GNU, GZIP жобаларындағы Zip бағдарламасы жиі қолданылады. бұл файл атауына .gz кеңейтуін енгізеді. Мысалы, бір кішігірім мұрағат bundle.tar.gz. пайда болуы мүмкін. Пайдаланушы Gunzip көмегімен тығыз орналасқан файлдарды ашып қарауы керек. Бұл мұрағатты не буманы қалпына келтіреді. Келесі қадам қара майды қолдана отырып буманы ашу болып келеді.

TAR файлын құру кезінде бастапқы файл (ы) мен параметрлері көрсетілген гудрон файлының атауы қалай көрсетілу керек, яғни сіз қара-

май файлын (с) құрастырасыз және мекенжай жолақша орнына файл (е) болып табылады. Бұл бұйрық келесі түрде көрінеді: қарамай -cfbundle1.tar * .txt Немесе ол келесі түрде бола алады: қарамай -cfbundle2.tar/TEMP. Бірінші жағдайда all.txt файлдарды қабылдап, оларды мұрағатқа өшіретке қояды, бұл уақытта соңғы өлшем барлық файлдарды тізімдемеде қабылдайды.

/ Температуралар, сонын ішінде тізімдеменің өзінің температурасы, оларды мұрағатқа қояды. TAR бұйрығы, файлды ашу -xfbundle2.tar. X параметрі файлды емдейтін экстракті белгілейді. bundle2.tar ашу арқылы тізімдемені жасауға болады, ал кейін сақталған мазмұнды жаңа тізімдемеге / TEMP сол жаңаға / орналастыру қажет.

Бағдарлама екіге бөлінгеннен кейін сіз оны жинақтауыңыз керек. Ең ашық бастапқы код бағдарламасы C немесе C ++ түрінде жазылған. C және C++ үшін Linux-та жиі қолданылатын құрылым НКУ (GNU жобасындағы C / C ++ компилятор) (p) және g ++ (GCC нақты бір параметрлерді орындау C ++) болып табылады. Өкінішке орай, C не C ++ бағдарламаларын жинақтаудағы тақырыптар файлдары (.h), бастапқы код файлдары (.c не .cpp), объективті файлдар (.o) шешім қабылдауда қолданылады. Linux процесті қарапайым қылатын үш пайдалы бұйрыққа ие. Бұл бұйрықтар орнатуды жасайды, бір арнаға келтіреді.

Конфигурациялау бұйрық сценарий жинақтарындағы файлдарды өзгерту және құру үшін қолданылады. Makefile сценарийі программистің бастапқы кодты жинақтауының нұсқамасы болып келеді. Бұйрық бір қалыпқа келтіре алады, мысалы бағдарламалық жасақтамаға енетін және жинақталуы тиіс үлгілерді көрсету әрі түрлі бағдарламалар файлын жасауда қолданылатын RIES директориясы үшін жүйелі әкімшілікті қолдану. Makefile мүлдем болмаса, онда конфигуралық бұйрық орындалуы тиіс. Ал егер Makefile бар болса, онда конфигуралық бұйрықтың болуы міндетті емес. Ол программист ойлап тапқан сценарий болып келеді. Конфигурация скрипін қосу үшін біреуі ./configure (одан көп емес)./ , ./configure (одан көп емес)./ кіреді, ол "бұл скриптті орындау" дегенді білдіреді. Соңына жеткеннен кейін енді Makefile қарастырамыз.

Makefile сценарийін орындаған кездегі бұйрық (ескертпе: Makefile не Makefile қайсысын тапса соны орындауы тиіс). Makefile сценарийі бағдарламалық жасақтаманы жинақтау үшін барлық бұйрықтарды жинайды. Makefile сценарийі компиляция нұсқаулығынан басқа бөлімдерден тұрады. Олар өздеріне орналастыру кезеңдерін орындау үшін орналастыру бөлімін (егер бағдарламалық жасақтама бір не бірнеше тізімдемелерде орналасқан жағдайда), тазалау бөлімін (керек емес уақытша құрылған файлдарды және қарамай ауқымын қосылған файлдарды

сақтай отырып тазалауға) енгізеді. Барлығы жақсы болған жағдайда пайдаланушы ./configure бұйрығын енгізу үшін барлығын жасауы тиіс. Жақсы Makefile өзін орнату бойынша орындаудан кейін тазарту жүргізеді деп атайды. Бұл қадамдар бірнеше секундтардан бірнеше минуттарға дейін созылуы мүмкін.

Бағдарламалық жасақтаманы орнату және жинақтауды орындауға тырысқанмен, әкімшілік тараулармен бөлінетін README мәтіндік файлды оқып шығуы тиіс. Makefile сценарийі, README орналастыру бойынша нұсқаулықты түсіндіретін программистпен жазылған. README файлын оқу әкімгер үшін жинақтау және бөлшектеу кезеңдерін орындауға көмектеседі. Әкімгер болып табылатын программист Linux-тің мақсатты жүйесіне бірнеше болжамдарды жасап қойған еді. Мысалы, программист GCC-тің нақты бір жерде орналасатының немесе /usr/ бен қолжетімділігін болжайды. README файлдары Makefile скрипті жұмыс істемеген жағдайда қателерді түзету үшін әкімгерге ескертпе тастауды өз мойнына алады.

13.3 суретте Makefile сценарийіне мысал келтірілген. Бұл қысқа мысал. Көптеген Makefile сценарийінің көлемдері үлкен болып келеді. 13.3 суретте Makefile сценарийіне келтірілген мысалда файлдағы мазмұнды жинақтауды түсіну жеңіл еместігі көрсетілген. Осы сценарийдің алғашқы төрт жолдары Makefile файлында қолданылатын ауыспалыны анықтайды. Бұл мысалда үш бағдарламалық файлдар бар: file1.c, file2.c и file3.c. Олар ауыспалы FILES-те сақталады. Келесі үш жол компиляцияның бастапқы кезеңінде қолданылатын ауыспалымен анықталады. Осы ауыспалы компилятор атауын (GCC, GNU жобасының C компиляторы), компиляция кезінде қолданылатын компилятор жалақшаларын, компиляцияны байлау кезеңінде қолданылатын құрылыс жалақшаларын анықтайды. Назар аударыңыз, файлдың оң жақ бөлігіндегі элементтер пікір қалдыру болып табылады (болашақта көрсетілетін таңбалар // бұрын болған). Сценарий орындалған кезде Баш интерпретаторы арқылы пікірлерге көңіл бөлінбейді.

Келесі алты жол бұйрық берілгенде сол бұйрықты орындау бұйрығын ұсынады. Осы алты желінің басты мақсаты әрбір бірінші желі дәлелдемемен бірге белгіні анықтайды (мысалы, \$ (FILES)), ал келесі бос орынды желілер бұйрықты орындайды. Барлығы: белгінің дәлелдемесі жоқ және ол файлдар астындағы бұйрықтарды орындайтын жалғыз бір топ болып келеді: белгі. Ол бұйрық үшін барлығын жасаймыз атты make-файлда жиі қолданады. Әр кезде топты байланыстырып қоятын НКҰ сұрақтары және барлық файлдар арасынан этикеткалы файлдарды ауыспалы \$ (FILES) (с.қ. үш C-файлдары) жағынан ажыратады. Әр

С бағдарламасын компиляциялау үшін астына с желілері қолданылады. қадамның үйлесімділік тәсілі мен компиляцияны бірге жасауға болады, алайда ыңғайлы болу үшін жиі бір-бірінен бөлінеді. Соңғы төрт желі міндетті түрде орындалатын нұсқаулыққа енеді. Бұрынғысы барлық уақытша файлдарды ағымдағы тізімдемеден тазартса, ал ақырғысы барлық сәйкес келетін бағдарламалық файлдарға қарамайды құрастырады. Бұл компиляцияның ең қарапайым жұмысы болғандықтан, секцияларды орналастырудың қажеттілігі шамалы. Программист Makefile файлдарын жазады (s) және оны барлық бастапқы кодымен әрі басқа да қажетті файлдармен (мысалы, құжаттар) жинастырады. Кейін әкімшілік Makefile сценарийін негізге ала отырып бағдарламалық жасақтаманы орналастырады. Сіздің сценарийді зерттеп өзгеріс енгізуіңіз сиректік емес. Мысалы, сызығының астынан Следующие шесть строк представляют собой команды для выполнения, когда команда выдается. Идея этих шести линий является то, что каждая первая определяет метку, возможно, с аргументами а на следующей строке, с отступом, команда для выполнения. Всего: метка не имеет аргументов, и его единственная команда, чтобы выполнить команду под файлы: метка. Это часто используется в для команды сделать все. Файлы этикетки перебирает все файлы в переменной \$ (FILES) (т.е. три С-файлы) и вопросы НКУ, связывающую команду на каждом. Под это линия с, которая используется для компиляции каждой программы С. и компоновку шагов можно сделать вместе, но часто разделенных для удобства. Последние четыре строки относятся к инструкциям, которые будут выполняться, если вопросы пользователей либо сделать чистыми или сделать деготь. Прежний очищает все временные файлы из текущего каталога, а последний создает деготь файл всех соответствующих программных файлов. Поскольку это был такой простой процесс, нет необходимости в установке секции. записывает файл и упаковывает его со всем исходным кодом и любые другие необходимые файлы. Затем администратор устанавливает программное обеспечение, опираясь на. Это не редкость, что вы, возможно, придется изучить сценарий и внести изменения. Например, представьте, что Makefile алған файлдарын тізімдемеге ауыстырды / USSR / жергілікті/ бен алайда сіз оны мында / USSR / бен ауыстырғыңыз келеді. Егер сіз компилятордың баптауларын жөнге от -g и -Wall тағы бір нәрсеге салғыңыз келсе.

Пакеттерді басқару құралдарын қолдану кезінде минут ішіндегі айналымды және нямды орналастыру, баптау және жасау неліктен маңызды? Олар ерекше кодты қолдану негізінде RPM веб-сайтындағы репозиторий арқылы қолданылады. Дегенмен олар екі жағдай негізінде жүзеге

асырылады. Біріншіден, кейбір ашық бастапқы код түріндегі бағдарламалық жасақтамалар орындалмаған формада келеді, сондықтан сізде оны өз қолыңызбен жинақтауыңызға тура келеді. Екіншіден, сіз немесе сіздің ұйымыңыздың бір тұлғасы бастапқы кодты өзгерткісі келсе, онда бәрі бірде Configure көмегі арқылы істеп және бұйрықты жасайды. Қолдануды қысқаша түрде келесі тарауда қарастырамыз.

Сервистік қызмет көрсету. Серверлер

Сервер термині үш түрлі заттарды белгілеу үшін қолданылады. Бірінші мамандандырылған компьютерге қызметті ұсыну үшін қолданылады. Серверлер ережеге сай жоғары өндірілген компьютерлерде қатты диск орналасқан. Кей жағдайларда серверлер шкафтарға құрылған, сондықтан бірнеше не ондық қатты дисктерді салуға болады. Сервер автономды компьютерге ұқсайды. Яғни, бұл біреудің үстел үстіндегі компьютері емес, пайдаланушылар да міндетті түрде серверге кіруі тиіс, егер ол сервер әкімшілігі болмаса.

Сервердің екінші термині компьютерде қосылатын бағдарламалық жасақтамаларда қолданылады. Серверлік бағдарламалық жасақтамалар өздерін мөр қызметтеріне, FTP-сервер, веб-сервер, сервер электронды пошта серверіне, мәліметтер қоры мен файлдық серверге жатқызады. Серверлік бағдарламалық жасақтамаларын орналастыру, құру мен басқаруда келесі қадамдар көрінеді. Мысалы, Apache веб-серверін құру мен басқаруда жұмыстың белсенділігі өздігінен артады.

Сервер үшін үшінші қолданыс бағдарламалық жасақтамаларының аппараттық құрылғысы мен серверді бірге құратын ақпараттар болып табылады. Мысалы, біз веб-серверге аппараттық құрал, бағдарламалық жасақтамалар және осында сақталатын құжаттар ретінде көңіл салуымызға болады.

Бұл тарауда бірнеше бағдарламалық серверлер түрлерін қарастырамыз. Apache веб-серверін минимальды конфигурацияда атап көрсетеміз. Төменде кеңінен тараған серверлер тізімі көрсетілген.

Веб - сервер

HTTP-сұранысына веб-сервердің жауап берудегі рөлі. . HTTP сұраныстары көбінесе веб-браузерлардан келеді, алайда басқа да бағдарламалар мен пайдаланушылар сұраныстарды жіберуге болады. HTTP гипермәгінді ауыстыру хаттамасы болып табылады. HTTP көптеген сұранымдары HTML-құжаттарына арналған (немесе ауыспалы- құмырсқа

HTML), алайда кез келген барлық веб-ресурстарда қолжетімді. HTTP FTP ұқсас келеді, өйткені файл сервис арқылы беріледі, бірақ оның FTP айырмашылығы пайдаланушы сервер атауын, не ресурстың атауын немесе жолын білмеуі тиіс. Олар барлығы URL ретінде танымал. Бастапқы ашық кодты Apache сервері бүгінде ең танымал веб-сервер болып келеді. Apache сұраныстарға қызмет көрсетумен қатар, динамикалық беттерді құруға арналған сценарийді орындайды, қауіпсіздік механизімін қолданады, сұраныстарды тіркейді, осының негізінде аналитиктар өздеріне қажетті пікірлерді іздестіре алады. Apache төменде жақсылап тұрып қарастырамыз. InternetInformationServer (InternetInformationServices деп аталады), не IIS, Microsoft веб-сервері болып табылады, тұңғыш рет WindowsXP операциялық жүйесі үшін қолжетімді, WindowsVista және Windows 7 жеткізіледі.

Прокси-сервер

Прокси-сервер ұйымда веб-парақшасының алып кәші ретінде қолданылады. Мақсаты мынада ұйымдағы көптеген адамдар бір ғана парақшаны қараса, онда осы парақшаларды кәштеу келешекте парақшаны локальды түрде қолдануына рұқсатын береді. Прокси-сервердің тағы бір қызметі белгіленген деңгейдегі анонимдікті қамтамасыз ету. Прокси-сервер қайтып келудегі белгілі бір контентті жою үшін қолданылады. мысалы, белгілі бір серверге шыққанда сұранымдарды тоқтарту (мысалы, Facebook) және нақтылы бір жауаптары бар сұрақтардан бас тарту (мысалы, "порно" сөзі). Squid Apache-нің жиі қолданылатын прокси-сервері болып табылады. Кальмар Apache сияқты бастапқы ашық код болып табылады

Дерекқор сервері

Дерекқор деректердің құрылымын сақтайтын қойма болып табылады. Дерекқорды басқару жүйесі бағдарламалық жасақтамалар болып табылады, ол пайдаланушылардың өңдеу, құру және іздеу сұранымдарына жауап береді. Дерекқор сервері желі бойынша дерекқорға қолжетімділікті қамтамасыз етеді. Сервер дерекқоры келесі міндеттерді деректерді талдау, деректерді өңдеу және мұрағаттауын орындайды. Дерекқор серверлерін бөліп, дерекқорды сақтау үшін түрлі механизмдері қолдануға болады, мысалы жүктеме тепе-теңдігін қолдау үшін желілерде ақпараттарды сақтайды. MySQL ең танымал дерекқор болып табылады. MySQLCommunity дерекқордың сервері болып келеді. GUI құрылы-

сын қолдау үшін жоғарғы жылдамдық пен құралды өзара байланыстыратын сервер кластері бар.

FTP-сервері веб-сервер сияқты файлдарды қабылдап оларды клиенттерге қолжетімді етеді. Веб-сервистан ерекшелігі кез келген жүктелген файлдар дискта сақталады, олардың көбі клиенттің веб-браузерінде сақталады. FTP пайдаланушының тіркелуін (HTTP ерекшеленгенде), не анонимді түрде қалуын қажет етеді. Анонимді пайдаланушы барлық қолжетімді файлдарды аша алады (/ паб атты арнайы тізімдемені). FTP HTTP ескі хаттамасы болып табылады. SFTP файлдардың ауысуын жүргізу үшін қауіпсіз амал болып танылғанмен, ол FTP хаттамасына негізделмеген. Сонымен қатар SSH қосуға болады, кейін FTP SSH ішінде қолдануға болады. FTP ең танымал құралы WS-FTP және FileZilla орындалуын қосады

Файлдық сервер

Файл-сервер файлдарды сақтайтын үлкен қатты дискі бар компьютерлерде болады. Ең қиын файлдық сервер ақпараттық файлды сақтап қана қоймайды, сонымен қатар желі бойынша қосылатын бағдарламалық жасақтамаларды сақтау үшін де қолданылады. Қолданылатын файлдық-серверлер компьютерлік желінің көзге көрінетін жерінде тұру үшін үлкен көлемді қатты диск қажет, алайда ол диск қымбат тұрады. Сондықтан 1 ТБ көлемді қатты дисктің құны 100 \$ құрайды, осының негізінде көптеген мекемелер файлдық-серверден бас тарты. Алайда файлдық-сервер бүгіндері көптеген мекемелерде қолданысқа ие, өйткені ол файлмен біріккен, тұтынушы файлдардың көшірмелерін лезде жасайды және жойылып кеткен файлдарға жол ашады. NFS қолжетімді көптеген түрлі файлдар бар (Network файлдық жүйесі), әлемдегі ең танымалдары Unix / Linux.

Электронды пошта сервері

Электронды пошта сервері электронды пошта қызметін ұсынады. Оның жұмысы, клиенттерден электронды пошта сұрауын қабылдайды және басқа электронды пошта серверлеріне хабарлар, сондай-ақ басқа серверлерден электронды пошта хабарламаларын қабылдап, пайдаланушыға электронды пошта келгенін ескертеді. Linux және Unix операциялық жүйелеріне Sendmail электронды пошта қызметі кіргізілген. Клиенттер электронды поштаға кіру үшін кез келген бағдарламаларды еркін пайда-

лана алады. Windows операциялық жүйесінде, Eudora, CleanMail және Microsoft Windows Server қоса алғанда, электронды поштаның сан алуан қолжетімді серверлері бар. Айырмашылығы FTP және веб-серверлер бар 3 - ITED саны хаттамаларын өңдеу үшін (бірінші кезекте, FTP, FTPS, HTTP және HTTPS), IMAP, POP3, SMTP, HTTP, MAPI және MIME өңдеуге тиіс электронды пошта серверлерінің хаттамалары.

Домендік атаулар жүйесі

Веб-сервер, FTP-серверлер мен электронды пошта серверлерін қолдану кезінде, ыңғайлы болу үшін, адамдар, әдетте, IP-лақап атты пайдаланады. Нақты IP-мекенжайлардан гөрі Алиасын есте сақтау оңай. Бірақ бағыттаулармен IP-лақапты пайдаланылуы мүмкін болмағандықтан, IP-лақап ат үшін IP-мекенжайларын ауыстыру талап етіледі. Бұл әдетте домен атаулары жүйесі (DNS) арқылы жүзеге асырылады. DNS карталау ақпаратынан тұрады, сондай-ақ басқа DNSS көрсеткіштерінен, сондықтан, егер сұрау салу мүмкін емес болмағанда, сұраным басқа DNS ауыстырылуы мүмкін. Linux жалпы бағдарлама DNS байланыстырады. Бағдарлама Dnsmasq форвардера бағдарламасы шағын Linux және (MacOS) желілерінде жиі пайдаланылады. Windows операциялық жүйесінде, Microsoft DNS Windows Server компоненті болып табылады. Cisco, сондай-ақ DNS-сервер-CiscoNetwork тіркеушіні (CNR) ұсынады. Біз осы тарауды Апат web-серверін Linux-қа қалай орнату және баптауды көрумен аяқтаймыз. Өйткені, Linux бағдарламалық жасақтамалардың басым бөлігінің бастапқы коды ашық болғандықтан, біз Configure көмегімен Apache қалай орнату, жасау, және орнату қадамдарын қарастырамыз.

Apache жүктеу үшін бірінші қадам. Сіз Apache httpd.apache.org ала аласыз. Apache Linux / Unix және Windows еркін қолданылғанымен, біз тек Linux-те қолданылғанын қарастырамыз. Жоғарыдағы аталған сайттан тек қана ең соңғы тұрақты нұскаудан Жүктеуді таңдаймыз (бета-нұсқаны жүктеп орналастырудың қажеті жоқ, өйткені Apache үнемі жаңаланып тұрады), Windows үшін екі формат және бастапқы код қолжетімді болып тұр. UnixSource таңдаулы элементтерінің бірін таңдаңыз. Шиферлеулер бар және олар қолжетімді. Мысалы, [HTTP-2.4.1.tar.gz](http://httpd.apache.org/dist/httpd/tar.gz) таңдағанда Apache 2.4.1 битондалған нұсқасын және Zipped таңдайды. Жүктеуден кейін оның бумасын шешіп мұрағаттау керек. Осының барлығын бір бұйрықпен орындауға болады:

```
деготь -xzfHTTPD-2.4.1.tar.gz
```

X, g және e нұсқалары "экстракт" стенді үшін, "Unzip" және "файл" дәлел ретінде берілген. Осы бұйрықтың негізінде ағымдағы тізімдемеден 2.4.1 жаңа тізімдеме шығады. Егер Ls директориясы негізінде орындаса, онда 13.4 суретінде көрсетілгендей тізімді көруге болады. Тізімдеме бірнеше тізімшелерден тұрады. Осы бағыттардың ең маңыздысы сервер болып келеді; барлық бастапқы кодты ұстанады C; өзіне түрлі файлдар тақырыптарын енгізеді C; Apache жеткізілетін түрлі үлгілерді құрастыру үшін қажетті кодты ұстанатын үлгіні с. Бұл тізімдемеде бірнеше сценарий бар. Build – конфигурация және баптау сценарийі Makefile сценарийін өзгертіп, Apache баптауға көмектеседі. Makefile.in және makefile.win сценарийі компиляциялау кезеңін орындауда қолданылады.

Әкімшілік жүйедегі келесі қадам осы қадамды орындау үшін бапталады. Осы бұйрықтардың орындалуы ./configure [KILT] ... [VAR = VALUE] / configure -h көмек аласыз. Ұзартылған хабарламаларды басып шығармау үшін «тыныштық режимді» қолдану керек. жалпы дәлелдемелер айналадағы ауыспалы жағдайларға байланысты белгіленеді. Олардың көбі тізімдемені немесе үлгіні өзгерту үшін қолданылады. кейбір келтірілген мысалдар:

--bindir = DIR -replace DIR орындалатын әдепкі тізімдеме
--sbindir = DIR -replace DIR орындалатын тізімдеме әдепкі жүйелік әкімшілендіру
--sysconfdir = DIR -replace тізімдеме әдепкі конфигурациялық с DIR
--datadir = DIR -replaceDIRмен әдепкі бойынша деректер тізімдемесі (әдетте / Var / WWW / HTML)
prefix = DIR -replaceDIRмен барлық әдепкі тізімдемелер
--enable-жүктеменің барлық модульдер -load барлық қол жетімді модульдер
--enable-модульдер = МОДУЛЬ-LIST -load барлық аталған модульдер,
--enable-AuthN-файл-load модульге аутентификацияны бақылауға қатысты
--enable-қамтиды -enable тарапқа сервер модулін қамтиды (CGI скриптті іске қосу үшін)
--enable-прокси -enable модуль прокси-сервер
--enable-SSL -enableSSL / TLS модуль

Қазіргі уақытта болуы мүмкін типтік команда:

`./configure --prefix = /usr / жергілікті / --enable-модульдер = ...`

Қажетті үлгілерді қай жерге толтыра аласыз бірге ... не төмен -Enable-үлгілер = толығымен, егерде бастапқы үлгі болмаса. Конфигурациялау бұйрықты орындау үшін бірнеше минут кетеді. Аяқталғаннан кейін сіздің Makefile ашық. Make бұйрығын басыңыз. Міне барлығыда жасалып бітті. Apache-нің орнатылған нұсқасы бар, бірақ ол қосылған жоқ. Apache орнатудың басқа да жолдары бар. Мысалы, CentOS орналастыру дискінің көмегімен атқарушы нұсқаға қол жетеді. Сіз оны Қосымшалар мәзіріндегі Add / RemoveSoftware бұйрығы арқылы орналастыра аласыз. Сіздің қолыңызда орналастыру дискісі болуы тиіс.

Apache орналасқандықтан енді оны қалай қолданатынымызды қарастырамыз. Apache қосу үшін Apache қызметін қосу қажет. Екілік тізімдемені табу керек, ол мынау /usr / жергілікті / apache2 / бен. Сіздің серверіңіз жан-жақты болуы тиіс.

Жоғарыда сипатталған конфигурацияны пайдалана отырып, сіздің барлық конфигурациялық файлдарыңыз /usr / жергілікті / apache2 / conf. сақталатын болады.

conf. негізгі файл httpd.conf. болып табылады. Сіз бұл файлды одан арғы күйге келтіру үшін редакциялайсыз және өз серверіңізді біздің нақты қажеттіліктеріңіз үшін бейімдейсіз. Бұл файл конфигурацияның жалғыз-файлы емес. Осының орнына, ол конфигурацияның командаларын бірнеше файлдарға бөлу үшін, жалпы болып табылады. Include директива-сы конфигурацияның басқа да файлдарын жүктеу үшін пайдаланылады. Сіз httpd.conf файлынан, көптеген өтініштер, соның ішінде түсіндірілгенін таба аласыз.

Сіз серверді күйге келтіру үшін не жасай алатыныңызды көрейік. Сіз қосымша IP-мекенжайларды немесе Apache ден қоятын болатын порттарды орната аласыз. Сіз журналдар файлдарының орналасуын және қазіргі уақытта тіркелетінмен өзгерте аласыз. Сіз ./configure команда-сында көрсеткен түрлі модульдерді пайдалана аласыз. Сіз балалардың санын өзгертуді, Apache уылдырлықты жейтінін және HTTP-скрау салуларды өңдеу үшін фондық режимде жұмыс істеуіңізді жалғастыруды қалауыңыз мүмкін. Бұдан басқа, сіз түрлі контейнерлерді қоса аласыз. Контейнерлер файлдық кеңістіктің ішкі каталогына, сол бір есімді немесе нақты URL ажырататын файлдар топтарына қалай нақты қатына-суды сипаттайды. Әрбір ретте, сіз конфигурацияның файлын өзгерткен кезде, бұл өзгерістердің күшіне ену үшін. Apache қайта іске қосу қажет. Сіз немесе ./apachectl тоқтау және ./apachectl іске қосу немесе қайта іске қосу ./apachectl командасын бере аласыз (соңғы команда Apache аталған

сәтте кезде ғана істейтін болады).

Сіздің Linux компьютеріңізде сақталатын сіздің «веб-кеңістігіңіздің» орналасқан орны / `USR` / жергілікті / `apache2` / веб болуы тиіс. Бұл сіздің сіздің веб-құжаттарыңыздың көбісін орналастыратын орын. Десе де, сіз файлдық кеңістікті жақсы ұйымдастыру үшін ішкі каталогтарды жасауды қалауыңыз мүмкін. Сіз сонымен қатар, сіз, мысалы, каталогтар контейнерлерінің көмегімен, жүйеге авторландырылған кіруді талап ететін `HTTPS` астына белгілі шектелген файлдарды орналастыра алатындай, жеке ішкі каталогтардың Паритерінің қолжетімділігін бақылай аласыз. `Apache` конфигурациясының файлы комментарийлер мен директивалардан тұрады. Директивалардың комментарийлері, ағылшын сипаттамалары файлдарда өзгерістерді енгізу үшін пайдалы ұсыныстармен қатар. Директивалар үш типке бөлінеді. Біріншіден, серверлардың директивалары барлық серверге әсер етеді. Екіншіден, контейнерлердің директиваларының әсері, сервер сияқты нақты директорийлер мен файлдарды өңдейді. Және ақырында, өтініштер кіреді, конфигурацияның басқа файлдарын жүктеу үшін пайдаланылады.

Сервер директивалары барлық серверге әсер етеді. Төменде сервердің жеті директивалары келтірілген.

`ServerName www.myserver.com` Пайдаланушының Апач

`Timeout 120`

`MaxKeepAliveRequests 100`

Тыңдау 80

`DocumentRoot "/usr / жергілікті / apache2 / веб"`

`ServerName` оның көмегімен бұл машина ден қоятын болатын `IP`-лақап аты болып табылады. Пайдаланушы және топ бұл үдеріс Linux жұмыс істеу кезінде пайда болатын атауды айқындайды (мысалы, сіз `ps` командасын жасайтын кезде). `Timeout` сервер команданың тайм-аутын беруге дейін клиентпен қатынауға тырысуға тиісті секундларды білдіреді. `MaxKeepAliveRequests` сұрау салулардың санын белгілейді, бір қосылыс алдында жіберілуі мүмкін. Қосылыс жабылады және жаңасы клиентпен сервер арасында қосылыс орнатуы тиіс. Тыңдаңыз сервер тыңдайтын болатын, порттың (тардың) нөмірлері сәйкестендіреді. Тыңдаңыз сонымен қатар егер де сервер бірнеше `IP`-мекенжайлары бар компьютерде жұмыс істесе, `IP`-мекенжайларына рұқсат етеді. `DocumentRoot` веб-парақ серверінде каталогтың орналасуын сақтайды. Әдепкі қалпы бойынша бұл мекенжай үшін, әдетте,
/ `Var` / `WWW` / `html`; Десе де, біз бұл саты кезінде қайта жүктедік ./

configure, біз барлық Apache үшін файлдық жүйелерде орналасқан орнын анықтаған кезде.

Include директивасы басқа конфигурациялық файлдарды көрсетуге мүмкіндік береді. Мысалы,

Қосыңыз conf.d / *. Конф

Apache іске қосу сәтінде conf.d ішкі каталогында табылған барлық other.conf файлдарын жүктейтін болады. Бұл директиваларды әрбір файлды қысқалау және ықшамдау сақтай отырып, көптеген файлдарға бөлуге мүмкіндік береді. Бұл сонымен қатар Ғаламтор серверінің әкімшісіне санаттағы топ директиваларына және де, қандай директивалар жүктелгенін және соның ішінде, кейбір өтініштерді түсіндіре отырып, жүктелмеуге тиісті емес екенін шешуге мүмкіндік береді. CentOS Linux в нұсқасында бұл ішкі каталогта табылған екі .conf файл бар: біреуі прокси сервисті күйге келтіру және .conf. сәлемдесу файлы.

Директиваның соңғы типі аталатын контейнерлер директиваларының нақты класы. Контейнер директивалар ол үшін қолданылуға тиісті орналасқан орнын сипаттайды. Контейнерлердің бірнеше типі бар.

14. Бағдарламалау

АТ-әкімшілеріне Тапн өткізген жүйелерге қолдау көрсету үшін жиі скриптерді жазуға тура келеді. Бұл тарауда бағдарламалау міндеттері қаралады. Бағдарламалаумен таныс емес тұлғалар үшін бұл тарау жай ғана сценарийлердің сыртынан гөрі, көбірек қамтиды. Біріншіден, тарауда 7-тарауда алғашқы рет ұсынылған бағдарламалау тілдерінің типтері қаралады. Бұдан әрі, тарауда көптеген мысалдардың көмегімен, бағдарламалау тілдерінің көбісінде табылған бағдарламалау тілі нұсқаулықтарының типтері сипатталады: кіріс және шығыс нұсқаулықтар, меншіктеу операторлары, таңдау, итерация операторларының өтініштері, ішкі бағдарламалар. Осы енгізумен, тарау содан кейін Bash түсіндірушілердің және DOS тілінде сияқты, скриптингке бұрылады, көптеген мысалдармен ілеседі.

Аталған тараудың үйрету мақсаттары мыналар болып табылады:

- Жоғары деңгейдегі тіл және төменгі деңгейдегі бағдарламалау арасындағы айырмаларды сипаттаңыз.
- Бағдарламалау саласындағы жаңартпаларды зерттей отырып, бағдарламалау тілдерінің эволюциясын талқылаңыз.
- Бағдарламалау тілі нұсқаулықтарының типтерін енгізу.
- Bash және DOS тілдердің сыртында сияқты бағдарламалау мысалдары арқылы ұсыну.

Бұл тарауда біз бағдарламалауды қараймыз және сценарийдің сыртын және Bash пен DOS сыртында қалай жазуды қарастырамыз

Тілдердің түрлері

Оның машиналық тілінде компьютерге жазылған, компьютердің командаларын орындай алатын бағдарламалардың типтері ғана, Машиналық

тіл компьютердің процессордың командалар жинақтамасына тікелей сілтеме жасайтын, екілік жүйеде жазылған нұсқаулықтан тұрады. Машиналық тілде бағдарламаларды жазу аса күрделі міндет болып табылады. Жақсылыққа орай, бағдарламашылар аудармашылар тілдерінің бар болуы салдарынан тікелей машиналық тілде бағдарламаларды жазудан баяғыда аулақ болуда. Аудармашы тіл бұл бір бағдарламаны кіріс деректері ретінде қабылдайтын бағдарлама, сонымен қатар шығару ретінде машиналық тілде бағдарламаны жасайды. Әрбір аудармашы тіл бір нақты тілден аталған процесордың машиналық тіліне аударуға бейімделген. Мысалы, бір аудармашы Windows компьютері үшін С-қа бағдарламаны аудару үшін, екіншісі Sun жұмытық станциясы үшін С-қа бағдарламаны аудару үшін пайдаланылатын болады. Ұқсас түрде, COBOL (Кобол), Ада, немесе FORTRAN тілінде жазылған бағдарламаларды аудару үшін түрлі аудармашы бағдарламалары талап етілетін болады.

Әрбіреуі бағдарламау тілінің басөа класына арналған тіл аудармашыларының үш тип бар. Бұл тілдер кластары құрастыру тілдері болып табылады, өз бетінше толтырылған тілдердің жоғары деңгейі, және түсініктеме берілетін тілдердің жоғары деңгейі. Бұл тілдердің үш типі ассемблерлер, компиляторлар, және, тиісінше, аудармашылар деп аталады. Ассамблея тілдері 1950-шы жылдары машиналық тілден құтылу үшін құрал ретінде әзірленген. Ассамблер тілінде бағдарламаны машиналық тілден гөрі жазу жеңілдірек болғанына қарамастан, бағдарламашылардың көпшілігі құрастыру тілінен аулақ болады, себебі олар бағдарламалаудың машиналық тілі сияқты сондай күрделі болады. С қарапайым нұсқаулықты қарап шығайық.

($y = 0$; $y < p$; $y++$) үшін
 $a[y] = a[y] + 1$;

олардың әрбіреуін арттыратын, массив орындарында p бойынша итерацияны. Тілді құрастыру (немесе машина), бұл қарапайым нұсқаулық бірнеше жеке нұсқаулықтардан бастап бірнеше ондықтарына дейін құрайтын болады. Үздік тілдер пайда болған соң, бағдарламалау ассемблерінің тілін пайдалану ең тиімді атқарылатын кодтың ықтимал болуын талап ететін (мысалы ерте компьютерлік ойындардан бастап) бағдарламашылар үшін шектелген болатын. Бүгінде жоғары деңгейдегі бағдарламалау тілдерінің басымдылығы салдарынан бағдарлама ассемблерінің тілінде ешкім жаза қоймайды.

Жоғары деңгейдегі бағдарламалау тілдері, машиналық немесе ассем-

блере жазу рабайсыздығы салдарынан туындады. Айырманы бейнелеу үшін, 14.1-суретті қараңыз. Бұл суретте, бағдарлама үш тілде ұсынылған: машиналық тіл (ойлап шығарылған компьютер үшін)

```
1111 00000000000001 Load # 1 INT = 1, B = 5, c;  
0010 1010100110000 магазинді біршама уақыт (a < B)  
1111 00000000000101 Load # 5 {  
0010 1010100110001 магазин б c = c + 2;  
0001 1010100110000 Top: Жүктеу B = B - 1;  
0100 1010100110001 Subt B}  
1000 00000000000000 Skipcond 00 E ( "% d", c);  
1001 0001000101010 Көшу шығу  
0001 1010100110011 жүктемелер c  
1110 00000000000010 Қосу # 2  
0010 1010100110011 магазин c  
0001 1010100110001 жүктемелер б  
1101 00000000000001 Subt # 1  
0010 1010100110001 магазин б  
1001 0001000100010 Көшу топ  
0001 1010100110011 Шығу: Жүктеме c  
0101 00000000000000 ШЫҒЫС
```

14.1-сурет

Машиналық тілде жазылған қарапайым бағдарлама (шеткі сол жақ бағана), құрастыру Тіл (ортаңғы колонка) және C (шеткі оң жақ бағана).

MARIE аталады), ассемблерде (тағы да, MARIE-де), және жоғары деңгейлі C бағдарламалау тілі. C коды анағұрлым қысқарак болып табылатынына назар аударыңыз. Бұл есептелген шамалар сияқты көптеген міндеттер (мысалы, $a = B - 1$), машиналық және құрастыру тілдерінде бірнеше қадам талап ететіндіктен орын алады, бірақ-та біреуі жоғары деңгейдегі тілде. Сонымен қатар ассемблер тіліндегі әрбір нұсқаулықта машиналық тілде баламалы командасы бар болады, сондықтан бұл екі бағдарламада бірдей өлшем бар.

Жоғары деңгейдегі бағдарламалау тілдері екі типтің біреуіне кіреді, құрастырылған тілдер жіне түсіндірілетін тілдер. Екі жағдайда да, тіл аудармашысы бағдарламаны жоғары деңгейдегі тілдерден машиналық тілге аударуы тиіс. Құрастырылған тілде компилятор деп аталатын тіл аудармашысы бұдан кешірек уақытта іске қосылуы мүмкін execut-қа-

білетті бағдарламаны жасай отырып, барлық бағдарламаны бір уақытта аударады. Түсінік берілетін тілде, бағдарламалық код түсінік берілетін ортада жазылады. Аудармашы жуырда ғана енгізілген команданы қабылдайды, оны машиналық тілге аударады, және оны орындайды. Бұл бағдарламашыға кодты әзірлеу кезінде эксперимент жасауға мүмкіндік етеді.

Мысалы, егер бағдарламашы нақты команда нені жасай алатынына сенімді болмаса, бағдарламашы команданы енгізіп, оны тестілей алады. Егер де ол қажет болған нәрсені жасамаса, бағдарламашы кейін қайтадан тырысады. Бағдарламаның түсінік берілген үдерісі бағдарламалаудың өз бетінше толықтырылған үдерісінен қатты ерекшеленеді, өйткені мұның эксперименталды сипаты көбірек және бағдарламаның әзірлемелері бөлшектеп жүзеге асырылуы мүмкін. Бағдарламаның жасаушысы барлық бағдарлама оның ішінен құрастырылып, тестілеуден бұрын жазылуын жиі талап етеді. Егер ол қажетті нәрсені жасамаса, бағдарламаны қайта тырысады. Бағдарламаның түсінік берілген үдерісі бағдарламалаудың өз бетінше толтырылған үдерісінен әжептәуір ерекшеленеді, себебі мұның эксперименталды сипаты көбірек және де бағдарламаның әзірлемелері бөлшектеп жүзеге асырылуы мүмкін. Бағдарламаның жасаушысы бүкіл бағдарлама одан құрастырылып, тестілеуден бұрын жазылуын жиі талап етеді.

Жоғары деңгейдегі тілдердің көпшілігі жақында өткен заманға дейін құрастырылған санатқа түспеген. Сценарийлердің тілдеріне түсінік беріледі, және бағдарламашылардың көбі сценарийлердің тілін пайдаланады. Түсінік берілген бағдарлама қарапайым түрде және тәсілі жақсырақ болса да, түсінік берілген бағдарламаның бір үлкен кемшілігі бар. Бағдарламаның компиляциясынан кейін, тіл аудармасының шаруашылық жұмысы и-мен аяқталды және, демек, бағдарламаны орындау аударманың қосымша уақытын емес, орындау уақытын ғана талап етеді. Түсініктеме берілетін бағдарламаны орындау үшін, әрбір команда ең алдымен таратылады, ал сосын орындалады. Бұл ұқсас өз бетінше толтырылған бағдарламадан гөрі, көбірек уақыт болу үшін, сондай орындау уақытына әкеп соқтырады. Десе де, көптеген заманауи түсінік берілетін тілдерде міндетті емес компиляторлары бар, аяқталудан соң, бағдарлама қалауы кезінде құрастырылуы мүмкін. жалпыланатын және түсінік берілетін бағдарламалау арасында туындаған басқа мәселе бағдарламалық жасақтаманы тестілеу болып табылады. Көптеген тіл компиляторлары бағдарламаның өз бетінше толтырылу кезінде қателердің көптеген типін табатын болады. Олар синтаксистік қателер ретінде белгілі.

Бағдарламашы ретінде (соның ішінде, инженер-бағдарламашы), қателер бар бағдарламалық жасақтаманы шығарылуын қаламаушы едік. Компилятормен ұсталатын қателер типтерінің бірі типтер сәйкессіздігінің қатесі. Қателердің бұл типінде, бағдарлама бұл типті сақтау жағдайында болмауға типіті ауыспалыда шаманың типін сақтауға тырысады, немесе нұсқаулық осы түрдегі операцияға жол бермейтін ауыспалымен операцияны орындауға тырысуы мүмкін. Екі қарапайым мысал тұтас ауыспалы шамада ондық үтір бар бөлшек шаманы сақтауға тырысады және жолдың ауыспалы шамасына көбейтуді орындауға тырысады.

Бұл қателерді компилятор байланыстырып бөгейді, сол уақытта СОМ бағдарламасы свалили. Түсінік берілетін бағдарламада, ол орындалғанша, бағдарлама таратылмайды, және сондықтан түсіндіруші бұл проблемадан аулақ болу үшін, тым кеш болған кезде, қатені орындау (орындау уақытының қатесі) кезінде ұстайтын ғана болады. Пайдаланушы, бағдарламаны іске қосып, түсіндірілген, барлық бағдарламалар неліктен теріс жұмыс істейді деген сұрақ қоға болады.

Орындау кезіндегі қате бағдарламаны орындау кезінде туындайтын сол болып табылады. Орындау кезіндегі қате компьютер қол астындағы міндетті мүлде орындамаған себебімен орын алады. Орындау уақыты қатесінің таралған нысаны арифметикалық шамадан тыс жүктеу болып табылады. Бұл есептеу жадтың көрсетілген орнына орналастыру үшін аса үлкен шамаға әкеп соқтырған кезде орын алады. Уақыт қатесінің басқа типі пайдаланушы орынсыз шаманы енгізген кезде туындайды, мысалы, сан сұратылған кезде, атауды енгізу.

Компилятормен ұсталуы мүмкін жоғарыда сипатталған, типтер сәйкессіздігінің қатесі. Бірақ егер де тіл компиляцияны талап етпесе, онда қате нақты команда орындалатын кезде ғана ұсталынады, және, сөйтіп, бұл орындау кезінде орын алады.

Қателердің үшінші саны логикалық қате болып табылады. Қателердің бұл типі бағдарламашының логикасы дұрыс емес болғандықтан туындайды. Бұл бірдеңе тривиалды болуы мүмкін, екі санды азайту сияқты, бағдарламаны екі санды қосу үшін арналған кезде. Ол сондай-ақ аса күрделі код нәтижесінде туындауы мүмкін, себебі бағдарламашы барлық ықтималды нәтижелерді қадағалай алмады.

Кодтан логикалық қатені қаншалықты қиын екенін бейнелейтін мысалды қарап шығайық. С-тен кодтың келесі фрагменті С (немесе С ++ немесе Java).

INT x = 5; қалқу у;
 $y = 1 / x;$

Код әуелі екі ауыспалыны жариялайды, бүтін x және қалтқының у нүктесін (шынайы). Ауыспалы x 5 нөмірімен инициализацияланады. Үшінші нұсқаулық $1 / x$ есетейді және нәтижесін у-де сақтайды. Нәтиже у-да сақталатын $1/5$ (0,2) шама болуы тиіс. Бұл компьютер үшін проблема болмауы тиіс, себебі у ондық нүктесі бар шаманы сақтай алатынын (x-тен бөлек) білдіретін, қалтқы болып табылады. Десе де, бұл нұсқаулықпен болған проблема C-те, C++ и Java, егер алым мен бөлгіш бүтін сандар болып табылса, бөлу бүтін бөлімше болып саналады. Бұл фактор бүтін сан болып табылатынын білдіреді, сондай-ақ. 1-ді 5-тен бөлу $1/5$ қалдықпен 0 шаманы береді. Бүтін фактор 0ге тең. Сөйтіп, $y = 1 / x;$ $y = 0$ келтіреді. Қалтқыда шама 0-ден 0,0-ге дейін түрлендірілу себебімен. Сөйтіп, у 0,2 орнына 0,0 болады! чумы логикалық қателер әрбір бағдарламашы дерлік.

Компьютерлік жабдық және операциялық жүйелері сияқты бағдарламалаудың жоғары деңгейлі тілдерін қысқаша зерттеуде өз тарихы бар. Мұнда біз бағдарламалау тілдерінің эволюциясын қысқаша қарап шығамыз, concentrat- ING бірінші кезекте соңғы 50 жыл ішінде жоғары деңгейдегі бағдарламалаудың көпшілікке мәлім тілдерінде.

Ең ерте тілдер машиналардың туған тілдері болды. Мұндай тіл бізге мүлдем бөтен болып көрінеді, сондықтан нұсқаулықтарда тек 1S және 0s бинарлық бар болады, немесе, бәлкім, нұсқаулықтар және деректер біршама жеңілдірек оқылуы үшін сегіздік немесе оналтылық түсінікте түрлендірілетін болады. Кез келген жағдайда, компьютерлік ғылымдарды немесе компьютерлік инженерияны зерттеген студенттерден басқа, бірнеше онжылдық ішінде машиналық тілді ешкім пайдаланбаған. 1950-ші жылдардың ортасында құрастыру тілдері енгізілді. Машиналық тілдер жағдайында сияқты, құрастыру тілдері күрделі міндет болып табылады. 1950-шы жылдардың соңынан бастап бағдарламашылар компиляторларды әзірлеуге кірісті. Тіл компиляторлары машиналық тілден жоғары деңгейлі бағдарламалау тілдерін аударатын аудармашылар жасай алды. Компиляторларды әзірлеу жоғары деңгейдегі бағдарламалаудың бірінші тілдерінің дамуымен қол ұстасып келді. Ең ерте жоғары деңгейлі тілді, кіру сигналы мен шығу операторымен бірге бірқатар бағдарламаны математикалық формулалар түрінде (осыдан тілдің атауы) көрсету үшін құралдың математикалық ғылыми есептермен істес болған бағдарламашыларға рұқсат етуге тырысты. Бағдарламашылардың көпшілігі компилятор тиімді кодты өндіреді дегенге скептикалық күйде болған, бірақ

бұл көрсетілгендей, компиляторла адаммен өндірілгеннен гөрі соншалықты тиімдірек болатын машиналық кодты тудыра алады, FORTRAN аса танымал болды.

FORTRAN, жоғары деңгейдегі бірінші тіл бола тұра, ҚҰРАСТЫРУ тілінде анықталған көптеген ерекшеліктері бар, және бұдан кештеу тілдерде табылған функциялардың көбісі жоқ болады. Мысалдардың біреуі Fortran-ның мәтінді терудің айқын емес ауыспалысына тәуелділік болып табылады.

Бағдарламашы тип бойынша ауыспалыны жариялағанын талап етудің орнына, ауыспалының типі ауыспалының атауына негізделетін болады. I бастап N дейін кез келген әріппен басталатын атауы бар ауыспалы бүтін болар еді, ал кез келген басқа ауыспалы шынайы болар еді (ондық нүктесі бар сан). Бұдан басқа, FORTRAN символдардың типі жоқ, сіз сандарды ғана, ал жолдарды сақтай алмайтындай. Ауыспалылардың атаулары ұзындыққа алты символмен шектелген. Сіз осылайша, income_tax ауыспалыны атай алмаушы едіңіз.

FORTRAN пайдалы болмайды, егер-өтініш, операторлардың GO TO пайдалануына осының орнына сүйенсе (тармактанудың шартсыз нұсқаулығының типі). Шартсыз тармақтар машина жасауда және бағдарлама тілін құрастыруда кеңінен пайдаланылады. Фортранда GO TO бағдарламашыға кез келген орыннан басқа орынға бағдарламаның басқаруын табыстауға мүмкіндік етеді. Біз осының мысалын осы бөлімде төменде қарап шығамыз. GO TO салдарынан, ал пайдалы емес, егер-өтініш, FORTRAN кодта осы қажет болғандықтан анағұрлым күрделі болатын логика бар. Өзінің кемшіліктеріне қарамастан FORTRAN ғылыми бағдарламалау үшін аса танымал болды өйткені бұл құрастыру мен машиналық тілмен салыстырғанда алға жылжудың зор қадамы. FORTRAN дамып жатқандай, Қорғаныс министрлігі COBOL (Кобол) бағдарламалаудың бизнес-бағдарланған тілді жасау үшін бизнес-бағдарламашылармен бірге жұмыс істеуге кірісті.

FORTRAN математикалық нотация сияқты сол түрде болған уақытта, өкімдерді құрайтын сөздер COBOL құқықтық синтаксисте жазылған COBOL құқықтық өтініштер болуға тиісті болса да, ағылшын сөйлемдері мен абзацтарында COBOL сөйлемі болды.

COBOL бағдарламасының идеясы әрбіреуінде өзінің меншікті тармағы болатын шағын процедураларды жазуы тиіс. Ішкі бағдарламаларды басқа ішкі бағдарламалармен шақырылатын болады, бағдарламалардың деректері деректер дерекқордан түсетін және шығу басқа дерекқорда орналасу жағдайында егжей-тегжейлі сипатталады. Сөйтіп, COBOL бағдарламалаудың көбірек бөлігі деректердің қозғалысы мен өңдеуі

болды, оның ішінде қарапайым математикалық есептеу мен сұрыптау. COBOL FORTRAN кейін бірнеше жылдан соң шығарылды, және FORTRAN сияқты табысты болды, алайда Cobol жағдайында, ол бизнес-бағдарламалауда табысты болды.

COBOL сондай-ақ кешіректеу тілдерде қол жетімді болатын көптеген функцияла жетіспейді. Ең басты кемшіліктердің бірі барлық ауыспалылар жаһанды ауыспалылар екендігі. Бағдарламалар, әдетте, оларды процедуралар, функциялар немесе әдістер деп кейде аталатын кішірек ішкі бағдарламаларға бөлінеді. Әрбір процедурада өзінің меншікті жад кеңістігі мен ауыспалылары бар. Осылайша, бағдарлама басқа функциялар оларды өзгерте алатынына қам жемей, жергілікті ауыспалылармен айла-шарғы жасай алады. Жаһанды ауыспалылар бұл идеяны бұзады, өйткені ауыспалы бағдарламаның бір орнында белгілі, немесе барлық бағдарлама бойынша белгілі бір процедура. Бір процедурада ауыспалының өзгерісін жасаудың басқа ішкі бағдарламаларда күтпеген салдары болуы мүмкін.

COBOL басқа проблемалардан зардап шекті. Ертерек COBOL массив ұсғымын қоспады. Онда COBOL басында параметрлердің берілісі болған жоқ (бір жағынан барлық ауыспалылар жаһанды ауыспалы болатындықтан). FORTRAN сияқты, COBOL тармактанудың шартсыз нұсқаулықтарынан тәуелділігі болды. Бұдан басқа, COBOL өзінің есептеу қабілеттіктеріне қатысты шектелген болды. COBOL алайда, екі өте маңызды ұғым жасады. Біріншісі құрылымдалған деректер ұғымы, енді деректердің құрылымы ретінде аталады, ал екіншісі символдар жолын сақтау мүмкіндігі болды. Деректердің құрылымы бағдарламашыға деректердің түрлі типтерінен тұратын ауыспалыларды анықтауға мүмкіндік береді. Мысалы, біз студенттің есімінен, майор, GPA, барлығы кредиттік сағат алды, және ағымдағы курстар қабылданады дегеннен тұратын студенттің жазбасын анықтай аламыз. Деректердің әртүрлі түрлері деректердің әртүрлі типтерін пайдаланумен ұсынылған: жол, заттық сандар.

Жасанды интеллект саласындағы зерттеушілер 1950-шы жылдардың соңында және 1960-шы жылдардың басында өз тілдерін әзірлеуге кірісті. Зерттеушілер кез келген FORTRAN немесе COBOL қолжетімді емес функцияның тіліне шақырды: символдар, рекурсиялар және орналастырудың динамикалық жад тізімін өңдеу. Бірнеше тілдерге қарамастан, бұл LISP-тіл болды - Лисп үйреніп кетті., LISP + граммдар жасаған жоғарыда аталған ерекшеліктерге толықтыру ретінде жеке-жеке жазылуға болатын еді. LISP өте инновациялық болды және 1990-шы жылдар арқылы, түрлі нысандарда пайдаланылу жалғасты. Бұл үш тілге сүйене отырып, Алгол тілі (алгоритм тілі) FORTRAN, COBOL және

LISP үздік қасиеттерін жаңа тілге біріктіруге тырысты. Алгол ақыр соңында, Құрама Штаттарынан гөрі Еуропада кеңінен танымал болды және оны пайдалану басқа тілдермен салыстырғанда шектелген болды, түпкілігінде Алголдың жойылуына әкеледі. Десе де, Алгол жаңа тілдер әзірленгенше, маңызды рөл атақатын болады.

1960-шы жылдары бағдарламалау тілдері катиондар салаларын көптеген түрлі қолдану үшін әзірленген. Сондай-ақ, ескі тілдерді алмастыру үшін жаңа тілдер немесе ескі тілдер кірмейтін про- Vide объектілер әзірленген. Кейбір танымал тілдер 1960-шы жылдардың басында ішінде FORTRAN, COBOL және ALGOL үздік қасиеттері бар болу үшін, IBM әзірлеген тіл, PL / I болды, осы арада ерекшеліктерді өңдеу; SIMULA, әзірлеу үшін модельдеу тілі, SNOBOL (Ішекті Oriented және символдық тілде), сәйкестік жолы; және BCPL, C ерте нұсқасы сияқты жаңа ұғымдарды енгізу.

1960-шы жалдардың соңы, бағдарламашылар бұл ерте тілдер өтініштердің шартсыз тармақтануына аса қатты сүйенгенін түсінді. Тілдердің көбісінде, өтініште GO TO (немесе Goto) аталды. Шартсыз көшу бағдарламашыға ағымдағы орналасқан орыннан бағдарламадағы басқа орында көше алады.

Мысал ретінде ұзындығы 10 бет болатын бағдарламаны қарап шығайық. 1-суретте соңынан 3-беттегі нұсқаулыққа тармактанатын GO TO ілестіртін төрт нұсқаулық бар. Бұл сатыда егер-онда басқаша өтініштер ұстанатын бес нұсқаулық бар

Сонда GO TO оператор ескертпесі бар, бұл 6-беттегі нұсқаулыққа көшеді, және GO TO өтініші бар болғандықтан, 8-беттегі нұсқаулыққа көшеді. 6-бетте GO TO өтініштер кезеңділігінің үш нұсқаулығы бар, 2-бетте 8-бетте бес нұсқаулық бар. 1-бетте тармақтары бар GO TO өтінішінде ұсыныс бар. Және тағы басқа. Бағдарламаны реттеуді тырысу ол арқылы қадағалауды талап етеді. GO TO операторларымен бірге бағдарламаның көмегімен жол тарту түсінбестікке әкеп соқтырады. Із спагетти үйіндісіне ұқсай бастауы мүмкін. Осылайша, шартсыз, тармактанатын ұғымдар спагетти коды деп аталған нәрсені жасауы мүмкін.

Ерте FORTRAN GO TO өтініштерде құрастырылған If-өтінішті пайдаланды. Өтініштің негізгі нысаны number1 сызығы, number2 сызығы, номер3 сызығы (арифметикалық өрнек) болып табылады. Егер арифметикалық өрнекте теріс шамасы мағынасы болса, number1 жолына, әйтпесе, егер арифметикалық өрнек нөлге тең болса, number2 жолына көшіңіз, олай болмаса, number3 жолына көшіңіз. FORTRAN-да тым жеңіл өндірілетін, спагетти кодын бейнелейтін 14.2-суретті қараңыз. Екінші нұсқаулық I-J болып есептелетін, Fortran шартты операторының

мысалы болып табылады егер теріс болса, тармақтар 10-жолға, егер нөл болса, тармақтар 20-сызыққа, олай болмаған жағдайда 30-жолға көшу керек.

$I = 5$, $J = 15$, и $K = -1$ деп болжай отырып, сіз 14.2-суреттегі код арқылы қандай нұсқаулықтар және қандай тәртіпте орындалатынын қадағалай аласыз. Егер сіздің жауабыңыз жоқ деген болса, бұл осындай дәрежеде GO TO өтініші пайдаланатын код арқылы жол тартудың рабайсыздығы салдарынан болуы мүмкін емес. Жаңа тілдерді дамытатын бағдарламашылар шартты ауысу тым құрамдастырылмаған деп шешті. Олардың шешімі бұдан да құрамдастырылған басқару операторларын әзірлеуі тиіс. Бұл итерацияның шартты операторларына әкеп соқтырады (мысалы, ілмектер сол арада), ілмектерді есептеп (ілмектер үшін), және енгізілген, егер өтініштер басқаша. Құрамдастырылған бағдарламалау тұжырымдамасын қабылдаған екі тіл АЛГОЛ: С және Pascal тармақтары ретінде әзірленді. 15 жылдан астам, осы екі тіл бағдарламалауда (C) және құрылуда (Паскаль) кеңінен пайдаланылатын болады. 1980-шы жылдардың басында әзірленген, Ада басқа да түрлі тілдерде табылған ерекшеліктерді қоса алғанда, екі тілдің ерекшелігін өз мойнына алады, жаңа функцияларды енгізеді, және онжылдықтар бойы АҚШ өкіметімен ғана пайдаланылатын тіл болады.

1980 жылы бағдарламалаудың нысанға бағытталған бағдарламалауға жаңа тұжырымдама пайда болды. Бастапқыда Симула бір нұсқасында әзірленген, бірінші нысанға бағытталған бағдарламалау тілі студенттің диссертациялық зерттеуінің нәтижесі болған Smalltalk болды. Идеясы деректердің құрылымы шынайы әлемнің объектілерін модельдей алуда тұрады (автомобиль немесе операциялық жүйенің терезелері сияқты абстрактылы объектілер сияқты физикалық объектілер болса да). Бағдарлама бір бірімен өзара әрекеттесетін объектілерден құралуы мүмкін. мысалы, егер жұмыс үстелінің объектісін бағдарламалау қажет болса, жұмыс үстелі өзін жабу немесе өз өлшемін өзгерту үшін, өзін азайту үшін өзінің орнын ауыстыру үшін, терезеге хабарламаларды жібере алады. Объект объекті деректерінің құрылымы және басқа объектілер оған өте алатын хабарламаларды өңдеу үшін қажетті барлық код ретінде сипаттайтын кластың анықтамасында көрсетілетін болады. Бағдарламалау бір-бірін шікі бағдарламалар деп атайтын идеялардан, бір-бірінің тізіміне хабарламалардың жөнелту объектілерін бағдарламалаудан өзгереді. 1980-ші жылдардың соңында C C++ дейін және LISP Common Lisp дейін көтерілді, екі жаңа тіл нысанға бағытталған болып табылады. кейін Ада Ada 95 дейін көтерілді. Және сосын, C++ негізделі отырып, Java әзірленді. Java, сол арада, ол құрастырылған және түсіндірілген болған-

дықтан бірегей тіл болды. Бұл туралы төменде нақтырақ талқыланады. C ++ және Java сияқты, үлкен маңыздылықпен, 1990-шы жылдардан бастап барлық тілдер нысанға бағытталған болды. Басқа тұжырымдама осы кезең ішінде енгізілген визуалды бағдарламалау болды. Бағдарламалаудың визуалды тілінде графикалық интерфейсі бар бағдарламаларды көшіру және GUI өзін дамытуды құлату, ал осын пайдаланушы GUI құраушыларымен өзара әрекеттесетін кезде, операцияларды өңдеу үшін кодты «кадрдан тыс» бағдарламалау әдісінің көмегімен оңай жасалуы мүмкін. Мысалы, пайдаланушының графикалық интерфейсінде екі түймешік пен мәтін жолы бар олуы мүмкін. Бағдарламашы бос панельге түймешіктер мен мәтін жолын жай кірістіре отырып, пайдаланушының графикалық интерфейсін жасайды. Сосын бағдарламашы түймешіктерді басқан кезде, немесе мәтінді мәтін жолына енгізу кезде орын алуы тиісті әрекеттерді жүзеге асыруы тиіс.

1990-шы жылы Ғаламторды жекешелендіру басталды. Компанияларға жеке тұлғаларға өз үйлерінен немесе кеңселерінен Ғаламторға қолжетімділікті алу мүмкіндігін сатуға рұқсат етілді. Осымен және World Wide Web құрылуымен, бағдарламаланудағы жаңа тұжырымдама тұңғыш енгізілді. Бұрын сіздік компьютеріңізге бағдарламаны іске қосу үшін, сіз компьютерлік нұсқаны алуыңыз тиіс болатын. Бағдарламашыға, бәлкім, бағдарламалық жасақтама әрбір платформада қолжетімді болатындай, әрбір платформа үшін компиляторды табу тура келеді.

Егер бағдарламашы C ++ бағдарламаларды жазса, бағдарламашы Windows үшін, Macintosh үшін, жұмыс станцияларына Sun үшін және тағы басқалары Intel үшін Linux базасында IBM үшін мейнфреймдер жасауы тиіс болар еді.

Бұған қоса, бұл бағдарламашыдан бірнеше түрлі компиляторларды алу үшін, ол сондай-ақ бағдарламашыдан әрбір компиляторды пайдалану алдында, ERS жасаушылары біршама өзгеше синтаксисті күте алатын болғандықтан, кодты түрлендіруді талап етуі мүмкін.

Java бағдарламалау тілін әзірлеу кезінде, авторлар бір жаңашылдық жасауға тырысты. Java компиляторы Java бағдарламасын олар байт-код деп атаған аралық тілге тарататын болады. Байт-код компьютерде жұмыс ітемейді, себебі тіл өзі қандай машина болған емес. Бірақ байт-код әрбір платформаға тәуелді болмайды. Компилятор іске қосу үшін қандай платформа бағдарлама арналғанға тәуелсіз, бір байттың бағдарламалық кодын өндімейтін болады. бұдан әрі, Java авторлары бір платформаға біреуден бірқатар Java (JVM) виртуалды машиналар іске асырылған. Виртуалды машина бірінен соң бірі, бағдарламаның байт кодының әр-

бір командасын, оны платформаның машиналық тіліне түрлендіру, және оны орындау жауапкершілігін алатын еді. Осылайша, виртуалды машина аудармашы болып табылады. Бұрын бағдарламаны түсіндіру шынайы болатын бағдарламаны жасаудан гөрі тиміділігі анағұрлым аз екені мәлімделген. Десе де, байт-код түсіндірмелері байт-код бағдарламаның командасымен ішінара аударылған болғандықтан, өз бетінше толықтырылған бағдарлама жұмыс істетіндей, тиімді дерлік жасалуы мүмкін. Сөйтіп, бұл комбинация және бағдарламаның түсіндірмесі өз бетінше толықтырылған бағдарлама жұмыс істейтіндей, соншалықты тиімді дерлік. Осындай тәсілдеменің артықшылығы Java төзе алушылығы болып табылады; бағдарлама компилятормен өз бетінше толықтырылған Java барлық веб-браузерлерде дерлік іске асырылған, JVM. JVMs бар кез келген платформаларда жұмыс істей алады, сондықтан Java барлық веб-браузерлерге дерлік іске асырыла алынады, осылайша тіл болып табылып, әдетте веб-браузерлерде іске асырылатын бағдарламаларлы іске асыру үшін пайдаланылады (мұндай бағдарламалар жиі апплеттер деп аталады).

Тәуелсіз бағдарламалар платформаларына ұмтылыс Ғаламтордың танымалдығының өсуімен уақыт келе артуды жалғастырады. Бүгінде Microsoft-тан .net ("желінің нүктесі") платформаның бағдарламауы байт-кодқа кодтың бұл өз бетінше толтыруында Java байт-кодтарға ұқсас тәсілдемеге мүмкіндік береді, содан кейін түсіндірілуі мүмкін. .Net Бағдарламалау тілдеріне C # (C ++ және Java әртүрлілігі), ASP (Active Server Pages), Visual Basic (визуалды бағдарламалау тілі), C ++, және J ++, аталатын Java нұсқасы кіреді. .Net платформасы платформаның бір тілінде жазылған бұл кодта Ява платформасынан тәуелсіз жаратылыс шеңберінен шығады, тілдердің басқа біреуінде жазылған код пайдаланылуы мүмкін. Осылайша, бағдарлама енді бір тілде жазылмауы тиіс, бірақ оның орнына бұл тілдердің қатарында жазылған жеке класстардан жасады мүмкін. Тілдерге келешекте қандай жаңа мүмкіндіктер қосылатыны, немесе бағдарламалаудың келешегі бізді қайда жаңа тілдерге қатысты көшіретіні белгісіз.

Ұзақ мерзімді мақсаты табиғи тілдерін түсіну үшін компьютерлерді бағдарламалауда тұрады. Табиғи тіл, адамдар бір бірімен өзара әрекеттесу үшін пайдаланылады.

Егер компьютер ағылшын тілін түсіне білсе, онда бағдарламашы бағдарламалаудың қандай да бір тілі шегінде кодты жазу туралы аландамаушы еді, компьютердің пайдаланушысы Linux және DOS сияқты, операциялық жүйелердің командаларының артынан синтаксисті білуі тиіс еді. Табиғи тілдер, алайда, екі мағыналыққа толы, сондықтан табиғи тілді түсіну

үшін компьютерді бағдарламалау аса күрделі міндет болып табылады.

Нұсқаулықтардың типтері

Бағдарламалық кодта қандай тілде болғанына қарамастан, нұсқаулықтардың сол бір типтері бар. Нұсқаулықтардың бұл типтері түрлітілдердегі мысалдармен қатар аталған бөлімде сипатталған.

Кіру және шығу нұсқаулықтары

Енгізу енгізу құрылғыларынан (немесе файлдардан) шамаларды алу және жадтағы ауыспалыларға шамаларды сақтау үдерісі болып табылады. Шығару – бұл әріптік шамаларды және шығару құрылғыларынан (немес файлдардан) ауыспалылар шамаларын жөнелту үдерісі.

Кейбір тілдерде есепті енгізу/шығарудың екі жеке жиынтығы бар болады: стандартты енгізуді (пернетақта) және шығаруды (монитор) пайдаланатын, сондай-ақ афйлдарды пайдаланатын. Басқа тілдер енгізу немесе шығару операторы көз/тағайындау құрлғысын көрсетуін талап етеді. В С, С ++ и Java, мысалы, кіру файлды стандарттық енгізу, стандарттық шығару және шығу файлы үшін әртүрлі пайымдамалар болады. FORTRAN-да және Pascal-да, стандарттық кіру мен шығуда әдепкі қалпы бойынша есептіліктің кіру және шығу деректерімен, бірақ файлды көрсету үшін өзгертілуі мүмкін. Бұдан басқа, енгізу/шығару операторлары деректердің түрлі типтері үшін кейбір тілдерді пайдаланады. Ada-да жолдарды Бекіту, бүтін сандар үшін, сандар үшін және тағы сол сияқты үшін түрлі кіру менттер бар. Тілдердің көпшілігі бір операторда енгізу мен шығару үшін бірнеше ауыспалы параметрлерге мүмкіндік етеді.

14.3-суретте төрт тілден файлды енгіздің нұсқаулығы көрсетілген: FORTRAN, С, Java и Pascal. Әрбір жағдайда, х, у, г үш деректердің кодтық кірулері (бүйін және екі шынайы немесе құбылмалы нүктесі бар сандар) – input.dat. мәтінді файлдан. Кодтың әрбір жиынтығы жай енгізу командасының өзінен гөрі көбірек талап етеді.

FORTRAN коды біріншісі енгізу үшін мәтінді файлды ашу үшін ашық өтінішті талап етеді. Файл кіру өтінішінде пайдаланылатын 1 деген белгілемені алады, READ. READ Операторы айтарлықтай қарапайым, алайда (1, 50), файлға, тиісінше, нысананы көрсете отырып және жол форматында жүгініңіз.

FORMAT Операторы кіру қалай түсіндірілуі тиіс екенін сипаттайды. Бұл ретте алғашы екі символ бүтін ретінде (I), ал келесі алты символ сол бір форматы бар кеінгі басқа қалтқысы бар ондық нүктеден (F6.2)

он жаққа қарай екі цифр бар санның (шынайы) құбылмалы үтірі ретінде қаралады.

C жағдайында, бірінші оператор FILE типіне ауыспалыны жариялайды. Сосын файл FOPEN өтінішімен ашылды. "Г" файл мынадан есептелуі тиіс, алайда жазыламуын көрсетеді. "% d% F% F" белгілемесі кіру ондық (бүтін сан) шамасы және құбылмалы нүктесі бар екі шама ретінде қаралуы тиіс және C-те Scanf операторында ауыспалыларға дейін пайдаланылатын символ C өз функцияларының параметрлерін табыстау салдарынан талап етіледі.

Pascal тілінде онда мәтіндік файл мәтіннің типі ртеінде жариялануы тиіс. Сосын мәтіннің ауыспалысына файлдың аты беріледі және ашылады. Және ақыр соңында, енгізу шамалары, Паскаль, бұл төрт тілдің ішінен қуатты ең аз ең қарапайым болып табылады. Шын мәнінде, Паскаль салдарынан қарапайымдылықпен жасалған.

Java-да, кез келген кіру өңделуі тиіс, бұл ерекшеліктер өңдеушісі ретінде белгілі. Бұл үлгілі жүйелердің блоқы арқылы жасалады (алайда Java сізге бұл үшін басқа механизмдерді береді). Бұдан әрі, BufferedReader типтегі ауыспалы енгізуді орындау үшін қажетті. BufferedReader түскен кіру сигналы бір ұзақ жол ретінде өңделеді. Жол құраушы бөліктеріне (үш сан) бөлшектенуі тиіс. Жол токенизатор досыны жасайды. Бұдан басқа, токенизатор нәтижелері синтаксистік талдаудың пайымдамаларының көмегімен сандық шамаларға түрлендірілуі тиісті жолдар болып саналады. Егер де ерекшелік кіруді орындауға тырысу кезінде туындаса, блок ұстау орындалады, бірақ мұнда блок кстау ешнәрсе істемейді.

Меншіктеу операторлары

Меншіктеу операторы шаманы ауыспалыда сақтайды. Ауыспалы оператордың сол жағында, ал шама оператордың оң жағында орналасады. Екі бөліктің бөлінуі меншіктеу операторы болып табылады. Меншіктеу операторының типтік нысаны = немесе: = (Pascal және Ada). Шаманың оң жағында әріптік шама болмауы тиіс, бірақ жиі өрнектің кейбір нысандары. Өрнектердің үш жалпы нысаны болады: арифметикалық, реляциялық және Boolean. Арифметикалық өрнектердесандық шамалар, ауыспалы және * B – 3 сияқты арифметикалық операторлар бар. Бұл ретте, а мен B болжам бойынша, санлық шамаларды сақтайтын ауыспалылар болып табылады. Арифметикалық операторлар +, -, *, / және модуль бойынша (Қалған дер алумен бөлу), жиі % көмегімен белгіленеді. Реляциялық өрнектерді қатынас операторлары шамаларды салыстыру үшін пайдаланады (аздау, тең, гөрі көбірек, гөрі көбірек немесе теңдес, тең және т.б.) Булев өрнектерін AND, OR, NOT, XOR пайдаланады. Сонда

сонымен қатар у квадрат түбірін алу үшін SQRT (y) сияқты шамаларды қайтаратын функцияларды шақырулар және конкатенция сияқты сондай жолдарды, функцияларды пайдалана отырып, жол өрнектері бар болуы мүмкін. Функциялардың шақырулары ішкі бағдарламаларда және ішкі бағдарламаларға талқыланады.

Теңдік белгісі (=) меншіктеу үшін тиі пайдаланатын болғандықтан, теңдікті тексеру үшін бұл тілдің қандай символы пайдаланылады (яғни, «Ъ теңдес жасайды»)? Біз әдетте математикадағы теңдік белгісін пайдаланамыз. Егер бағдарламалау тілі екі мақсат үшін бір және сол бір символды пайдаланатын болса? Егер ия болса, онда бұл бағдарламашыны жаңылыстырып қоймай, компиляторды шатастыратын болады, сондай-ақ. Осының орнына меншіктеу мен теңдік үшін символдардың екі әртүрлі жиынтығын пайдаланыңыз. 14.1-кесте қатмаасыз етеді

14.1 Деректердің тағайындалуы және теңдік бірнеше тілдегі Операторлар

Тіл Меншіктеу операторы теңдік Операторы тең емес Оператор

Ада: == / =

C / C ++ / Java, Python, Ruby

=== (2 теңдік белгісі)! теңдік белгісі

COBOL беру Тең (немесе =) тең емес (немесе жоқ =)

FORTRAN = .EQ. .NE.

Паскаль: == <>

Perl = экв (жолдың), = (саны)

пе (жолдың), != (нөмірдің)

PL / I = <>

бірнеше тілдерде салыстыру. Ол сонымен қатар оператор «тең емес» екенін көрсетеді. PL / I меншіктеу мен теңдіктің екеуіне = пайдаланатын жалғыз тіл болып табылатынына назар аударыңыз.

Меншіктеу операторы тілдердің көбісінде бір ауыспалы шаманы менкітендіреді. Кейбір тілдер, алайда, бір командада бірнеше тағайындауға мүмкіндік етеді. В, С, мысалы, $x = y = g = 0$; 0-де барлық үш ауыспалыны белгілейді. Python-да сіз оларды x -ты 5-шамаға менкітейтін, сосын u -ды $x + 1$ (немесе 6) шамасына меншіктейтін, в x , $y = 5$, $x + 1$ сияқты,

үтірлері бар сол жақта оларды бөлшектей отырып, бірнеше ауыспалығы тағайындай аласыз.

Бұл құлақша бағдарламадағы командалардың санын азайтуы мүмкін, бірақ-та бағдарламаның жеңіл оқылуын төмендетуі мүмкін. С-та сондай-ақ қысқартылған операторлар бар. Оларға $x = x + 1$ сияқты болып табылатын және $x = x + y$ орнына $x += y$ жазуға мүмкіндік беретін ауыспалыны көбейту үшін ++ пайдалану жатады.

С сондай-ақ меншіктеу операторы бөлігінің өрнегі шегіндегі тағайындауға рұқсат етеді.

$a = (b = 5) * c$ пайымдамасы; біреуінде екі меншіктеу операторы. Біріншіден, $b = 5$ -ке тең болып белгіленеді, сосын b жаңа шаманы пайдаланумен $b * c$ тең болып белгіленеді. Егер тіл өз бетінше толықтырылса және ауыспалылар жарияланса, онда компилятор типтердің сәйкессіздігін тексере алады. Қатенің бұл типі егер шегіну туралы оң жақ сол жақтағы ауыспалымен сәйкес келмейтін шаманы тудыратын болса, пайда болады. Мысалы, егер x бүтін ауыспалы болса, онда меншіктеу операторы $x = 3,1415 * y$ типтердің сәйкессіздік қатесін шығарады, себебі сол жақ бөлікке шынайы (құбылмалы нүктесі бар) сан кіреді.

Бұл у-дың қандай типі бар болуына байланысты әділді.

таңдау өтініші

шарттар шындық және өтірік ретінде есептелетін өрнектер болып табылады. Шарттар бағдарлама одан әрі не істеу керектігін анықтау үшін бағдарламаларда пайдаланылады. Олар өтініштердің екі типінде пайдаланылады, есептіліктің таңдау және итерация (немесе ілмек) өтініштерінде (итерациялық есептеліктердеқаралатын). Шарттың идеясы бір немесе бірнеше ауыспалыда сақталатын шама салыстырылатында болып табылады. Салыстырулар реляциялық операторлаған негізделуі мүмкін. Сонымен қатар логикалық операторлар және арифметикалық операциялар пайдаланылуы мүмкін. Мысалы, шарт $x > y$ -дан көбірек бар ма немесе $x > y$ екенін тексере алады. Басқа шарт егер x у-ге тең немесе x г-ге тең болса, тексере алады. Түрлі тілдер реляциялық операторлар (аздау, гөрі көбірек, теі және т.б.) және логикалық операторлар үшін (және, немесе, жоқ) әртүрлі болатын символдарды пайдаланады. Таңдау операторы

Бірқатар нұсқаулық күйін бағалау негізінде орындалатыны немесе өткізілетіні қажет деп шешеді. Өтініштердің бұл типін егер-онда өтініш деп аталады (немесе жай егер өтініш). Егер-онда өтініште, егер шарт шынайы болса, онда өтініштің бөлігі (әдетте сосын тармақ деп аталатын) орындалады. Егер шарт жалған болса, онда тармақ өткізіліп жіберіледі.

Егер-онда-өтініште, еге гарт шынайы болса, онда ұсыныс орындалады, шарт жалған болса ескерту тағы орындалады. Сіз егер-онда-өтініш болса, сіз пайдаланасыз, егер шарт шынайы болса, сіз орындауға арналған кодты ғана қалайтынсыз. Бұл кейде бір жақты таңдау деп аталады. Сіз егер сіз ол қандай күйде болуына қарамастан, кодты орындау үшін қаласаңыз егер-онда-басқа өтініш пайдаланушы едіңіз.

14.4-суретте үш тілдегі кодтың мысалы бар: Pascal, C и FORTRAN IV (FORTRAN ерте нұсқасы). Әрбір нұсқада егер x y тең немесе x y теі болса x -ке 1-ді қосып, және олай болмаған жағдайда x -тен 1-ді шегеріп, сол бір міндетке ие болады. Паскаль кодын түсіну оңай. В C, теңдіктің екі белгісі оператордың және екі тік барының «теңдігін» құрайды (||) операторды құрайды NEMECЕ. FORTRAN коды GO TO есептілікке сүйенеді, интуитивтігі аз болып табылады. Код арқылы оқыңыз және егер сізосыны түсіне алсаңыз, қараңыз. C-те «сосын» деген сөз төменге түсірілгені және шарт жақшалардың ішіне орналастырылатыны байқалады.

Егер-онда-басқаша оператор кодтың екі жиынтығы, «бір зат немесе басқа» жағдайдың арасында таңдайтын болса. Оладың арасында таңдау үшін, кодтың жай екі ықтимал жиынтығынан гөрі онда көбірек болса, не істеу керек? Тілдердің көпшілігі егер-онда, және егер-онда тағы өтініштер сондағы бар болу үшін, онда тармақ немесе басқа-тармақ мүмкіндік етеді.

Бұл салынған деп аталатынды жасайды, егер-онда басқа өтініш.

14.5-суретте студенттің сандық класының негізінде тағайындау үшін, кластың қандай әріпі екенін қалай анықтауға болатынын көрсете отырып, Паскаль тілінде мысал көрсетілген. Біз типтік 90/80/70/60 сынамамен пайдаланамыз. Сәйкес келеді деген шартқа сүйене отырып, код ауыспалыға келесі шамалардың біреуінің әріпін береді: 'A', 'B', 'C', 'D' или 'F'.

Соңғы пайымдама егер идея келесіде болса талап етілмейтініне назар аударыңыз: бірінші шарт шынайы болса, онда бірінші тармақ онда орындалады және барлық салынған оператор аяқталады; олай болмаған жағдайда, бірінші тармақ әлі орындалады. Бұл бөлім тағы егер-онда-басқа өтініш бар. Бұл екінші шарт үшін, еге бұл рас болса, онда екінші тармақ, онда орындалады және оператор аяқталады, олай болмаған жағдайда, екінші тармақ тағы да орындалады. Бұл тармақта тағы сондай-ақ егер-онда-басқа өтініш бар. Егер тармақ (үшінші шартқа) шынайы екені туралы шарт орындалса, онда үшінші тармақ, онда орындалады және оператор аяқталады, олай болмаған жағдайда, үшінші тармақ тағы да орындалады. Үшінші бөлімде тағы бір өтініш бар. Егер бұл

төртінші шарт шынайы болса, онда төртінші тармақ, онда орындалады және оператор аяқталады; олай болмаған жағдайда, соңғы тармақ тағы да орындалады.

Бірнеше басқа ескертпелер мында ескерілу қажет. Тереңдету. Жоғарыда келтірілген кодты оқу үшін ыңғайлырақ жасауға көмектеседі, алайда Кодекстің немесе қол жеткізілген сипаттамалардың дәлдігіне әсер етпейді. Яғни, барлық кіріктірілген қосымша парақтарды компьютер елемейді. Бұдан басқа, капитализацияның маңызы жоқ. Және, ақыр соңында, сіз кодтың барлық жиынтығында соңында нүктелі үтір бар екенін байқай аласыз. Pascal және C, C ++ және Java арасындағы айырмалардың бірі өтініштердің соңына дейін Semicon пайдалану болып табылады. Pascal тілінде, барлық құрылымды аяқтау үшін біреу ғана пайдаланылады. Жоғарыда келтірілген мысалдарда онда қалған тармақтар да жеке операциялардан тұрғандарына назара аударыңыз. Мысалы, 14.4-суретте көрсетілген мысалда, әрбір сөйлем біреуін жасады: инкремент x немесе декремент x. Бұл жиі болатындай сияқты, ережеде бірнеше нұсқаулықтар болады. Pascal тілінде, мысалға, сізге кодты келесі түрде ие болуға болады:

Егер $x > y$
Онда $x := x + y$;
 $y := 0$;
д: $x := x + y$;

Шегерістер және басқа да бос орындарға назар салынбайды, себебі компилятор тармақты бір уақытта аяқтайтынын біледі? Яғни, д: $x + y$ барсондағы тармақтың бөлігі? Шын мәнінде, бұлолай емес, және бірде-біреуі $y := 0$ емес; тіпті егер аралық ол бар екенін түспалдаса да. Pascal және C сияқты сондай тілдерде компилятор бір командаға ие болу үшін ескертпені күтеді. Егер сіз бірнеше нұсқаулықтарды көрсеткіңіз келсе, оларды блокқа орналастру қажет. Блок бір тұтас ретінде қарауға қажетті нұсқаулықтардың жиынтығы білдіреді. Бұл сізге сол немесе басқа п бірнеше нұсқаулықтарға тие болуға мүмкіндік етеді. Паскальда блок басталады және аяқталады деген сөдердің көмегімен белгіленеді, C-да лок символдардың көмегімен белгіленеді. Жоғарыда көрсетілген өтініштің Pascal және C дұрыс нүктелері, егер-онда төменде көрсетілген. Сіз $y := 0$ нүктелі үтірмен аяқталмайтынын байқай аласыз. Паскаль үтір бар нүкте қағидалары біршама қиындау:

Егер $x > y$ Онда, егер $(x > y)$ {Бастау $X = Z$;

```
x: = г; y = 0;  
Y: = 0}  
Соңы; д = x + y;  
д: = x + y;
```

Ada сияқты кейбір тілдерде айқын ENDIF өтінішпен аяқталады. Басында FORTRAN оның орнына, сіз GO TO өтініші негізінде ережелердің соңын белгіледіңіз. Біз блоктың оралуын көреміз, біз өтініш итерациясына және ішкі бағдарламаларда және бағдарламаларға алған кезде. Төменде жоғарыдағы сол бір код келтірілген бірақ осы ретте Python жазылған. Python шаманы теңестіру кезінде бірегей тіл болып табылады. Бізге блоктың индикаторлары керек еместігіне (басталады немесе аяқталады немесе {және}) назар салыңыз, бірақ оның орнына шегерістер екі нұсқаулық егер $x > y$: соңғы тармақтың бөлігі болып табылғаннан соң көрсетеді. Соңғы пайымдама үшін шегеріссіз онда бұл ол сондағы тармақтың бөлігі болып табылмайтынын білдіреді.

```
Егер  $x > y$ :  
x = г y = 0  
д = x + y
```

if-басқа өтініш енгізілген басқа мысалды қарайық, осы ретте C-те. Сіз x, y, және г шамалары негізінде орындала алынатын үш әртүрлі ереже бар екенін көресіз, Көрсетілген x, y және г шамаларын сіз үш ереженің қайсысы орындайтынын анықтайтын күйде болуыңыз тиіс. Мысалы, егер $x = 5$, $y = 3$ және $г = 0$, тармақ нені орындайтын еді?

```
егер ( $x > 0$ )  
егер ( $y > 0$ )  
... // 1-тармақ, егер тағы ( $г > 0$ )  
... // 2-тармақ  
тағы ... // 3-тармақ
```

1-тармақ орындалатын болады, егер $x > 0$ және $y > 0$. 2-тармақ орындалмайтын болады егер $x > 0$, $y \leq 0$ және $г > 0$ (егер $x \leq 0$, біз екінші if-операторына ешқашан жетпейміз және сондықтан біз тағы if- өтінішке ешқашан жетпейміз, ал егер $y > 0$, онда біз басқасына ешқашан жетпейміз, егер-өтініш). 3-тармақ қашан орындау керек? Яғни, бұл ($г > 0$) шартымен қосақталған соңғы шарт немесе ($x > 0$)? Ол c-пен жұпта емес ($y >$

0), себебі басқаша, егер өтініште бұл шарт үшін ереже бар болса. Кодтың аталған мысалында, түсінікті емес, егер соңғысы біріншісімен әлі байланысты болса, егер немесе үшінші егер. Бұл жағдай, тағы қатып қалған ретінде белгілі. Егер біз белгілеу блокын пайдалануға тиісті болсақ, біз келесі әрекеттердің бірі ретінде бұл проблеманы оңай шеше алатынбыз:

```
егер (x > 0) {егер (x > 0)
егер (y > 0), егер (y > 0) {
... // 1-тармақ ... // 1-тармақ
} Тағы егер (г > 0)
басқаша, егер (г > 0) ... // 2-тармақ
... // 2-тармақ}
тағы... // 3-тармақ тағы ... // 3-тармақ
```

Кодта сол жақта, соңғысы екіншісімен бірге жүреді, егер өтініш ($y > 0$), ал сол жақтағы кодта, соңғысы біріншімен қоса жүреді, егер өтініш ($x > 0$). Десе де, C-та, біз белгілеу блокын пайдаланбауымыз тиіс, себебі, әдепкі қалпы бойынша, үзілгендердің кез келген басқа тармағы ең соңғы күйден үнемі біріктірілген. Сөйтіп, 3-тармақ орындалады, егер $x > 0$ және $y \leq 0$ және $г \leq 0$. Егер біз, әдепті қалпы бойынша, қолдану үшін (оң жақтан жоғарыдан оқиға) қаламайтын болсақ, біз сосын белгілеу блокын пайдаланушы едік. Түрлі тілдерде үзілгендерге немесе операторлар кезінде қатысты түрлі ережелері бар. Айтпақшы, егер $x = 5$, $y = 3$ және $г = 0$, 3-тармақ орындалатын болады.

Таңдаудың бірнеше операторларының басқа да ныандары бар. Паскальде олар оқиға өтніш деп аталады. C мен Java-да олар коммутатор операторлары деп аталады. Біз оларды, сіз оларды бағдарламалаудық кез келген кіріспе курсынан көруге қаласаңыз да, мында қарастырмаймыз.

Итерациялық өтініштер

Код жиынтығы бірнеше рет орындалу кездегі сәттер бар. Еңбек ақыны есептеу операциясын орындайтын еңбек ақысын есептеу бағдарламасын жазғыңыз келетініңізді мысал ретінде қарап шығайық.

Бағдарлама қызметкердің деректерін енгізуі, қызметкердің еңбек ақысын және салықтарын және есептеуі, нәтижелері шығаруы, сонымен қатар жапаар жерде қорытындыларды қолдауы тиіс. Сіз кіру қадамдарын, есептеуді және шығуларды орындайтын, сосын аяқталатын кодтың тікелейжелісі түріндегі бағдарламаны жазуды ойластыра аласыз. Бұл бір

қызметкердің еңек ақысын есептеуі болатын еді. Сіз бір қызметкерге бір рет бағдарламаны іске қосуды қаламайтын едіңіз, себебі бұл бағдарламаны іске қосу сізді ашуландырушы еді. Бұдан басқа, сіз барлық қызметкерлердің қорытынды деректерін есептей алмайсыз. Онын орнына «тікелей желі» кодын жауға және оны итерация нұқаулығына орналастыруға болатын еді. Итерация операторы код жиынтығын қайталау үшін пайдаланылады.

Өтініш итерациясының екі негізгі түрі бар. Шарт шындық ретінде есептелетін болғанша, шартты, немесе логикалық, цикл кодты қайталайды. Есптеу циклы (ілмек ретінде жиі белгілі болатын) бірнеше рет немесе бастапқы және соңғы нүкте негізінде (мысалы, 1-ден 10-ға дейін), немесе тізімде сақталған шамаларды іріктейді. Ілмекті есептеу кейбір кезде ілмектер үшін деп аталатын себебі тілдердің көбісі бұл есептеу циклы екенін көрсету үшін, резервіленген сөзі пайдаланатынын болып табылады. Төменде Паскаль кодтың үлгілері келтірілген. Біріншіден, біз логикалық циклды көреміз.

```
Сол арада  $x > y$   
бастау  
...  
соңы;
```

Бұрынғы циклда, шарт $x > y$. Бұл цикл орындауды жалғастыратынын білдіреді, сол арада $x \leq y$ кбірек қалады. Бастау ...соңы блокын пайдалануға назар аударыңыз. Егер-онда және егер-онда басқаша сияқты пайымдамалар жағдайында, егер код циклдың сол уақыты (циклды денесі ретінде белгілі) бір командадан көп командадан тұрса, блокты пайдалану талап етіледі. Жоғарыда келтірілген циклда, $y \leq x$ -тен гөрі көбірек аз емес бола салысымен, цикл аяқталады және бағдарлама өтініш аяқталғанға дейін келесі нұсқаулықтан басталады. В. С, сол бір цикл келесі түрде көрсетіледі:

```
сол арада ( $x > y$ ) {  
...  
}
```

Контур үшін екі тәсілдің біреуінде пайдалануы мүмкін. Біріншіден, цикл бастапқы шама және бірліктермен аяқталатын шамасы ретінде белгіленетін, қадам өлшемі аталатын бірқатар шамалар арқылы қайта-

ланады. Мысалы, біз 2s-те 1-ден 11-ге дейін циклды көре алатын едік (цикл итерацияны 1, 3, 5, 7, 9, 11 есептеу жолымен жасайды). Екіншіден, цикл тізім бойынша іріктеледі. Мысалы, егер ауыспалы шамалар итізімін сақтаса, айталық, $x = \{1, 10, 25, 39, 44\}$, онда цикл бес рет қайталаанады, тізімдегі бір шамаға бір рет. Кез келген жағдайда, циклде циклдың индексі бар. Бұл ауыспалы қазіргі уақытта итерация болатын шаманы сақтайды.

Есептеу циклында, индекс әрбір 1, 3, 5, 7, 9, 11 шамасы бойынша қабылданады, сонымен қатар циклдың тізімінде, ол тізімнің әрбір шамасын қабылдайды (1, 10, 25, 39, 44). Цикл есептеу циклынан төмен. Циклдың индексі I бірінші итерация ішінде 1 шаманы қабылдайды 2 - екінші итерацияда, 3 -үшінші кезінде, және оған жеткенше әрі қарай 10, оның соңғы итерациясы.

I үшін: = 1-ден 10-ға дейін жасау

...

соңы;

Циклдың индексі циклдың денесіне сілтеме жасай алады. Сіз бұл шаманы есептеулердің бөлігі ретінде пайдалана аласыз. Келесі код 1-ден 10-ға дейін іріктеу ғана емес, ағымдағы сомада циклдың ағымдағы индексін қосады. Сосын бұл код 1-ден 10-ға дейінгі шамаларды жинақтайды (яғни, ол $1 + 2 + 3 + \dots + 10$ есептейді). Бұл жинақтау циклы ретінде белгілі.

Сума: = 0;

I үшін: = 1-ден 10-ға дейін жасау

Сума: = Sum + I;

Бізге циклдың денесін бастау ... өтініштің соңы ішінде орнын ауыстыру керек емес екеніне назара аударыңыз, өйткені циклдың денесі бір командадан тұрады. Сол бір код төменде C-те келтірілген. C цикл үшін шамалы қарапайым Pascal циклмен салыстырғанда, түсініксіз көрінеді. В C, жақшалардағы бірінші тармақ циклдың кез келген индексін (терін) инициализациялайды, екіншіде қосулы Aco шарты бар, ал үшінші тармақ индекс шамасының үдею немесе азаю қадамын орындайды.

сума = 0; (я = 0; r <= 10; I = I + 1) үшін

сума = сума + I;

С шын мәнінде бірқатар затбелгіні ұсынады. Олардың біреуі $I = I + 1$ орнына пайдалана алынатын $I++$ пайымдама болып табылады. Басқа сома $= \text{сумма} + I$ сомасының орнынан пайдалана алынатын $+= I$. Төменде ($I = 10; > 0; i--$) үшін жоғары емес, төмен іріктелетін болатын цикл үшін келтірілген (циклдың операторы үшін, циклдың денесі үшін емес).

С пен Pascal арасындағы айырмалардың бірі С өз қадамының өлшемін түрлендіре алатыны болып табылады (мысалы, $2s$ немесе $5s$ бойынша итерацияны), сол арада Паскаль жоғаырға немесе төмен итерацияны ғана істей алады.

Кейбір тілдер екі шарттының және ілмектерді есептеудің түрлі нысандарын ұсынады. С мен Java ілмектер сияқты сол арада жасайды. Олардың арасындағы айырма шарт қандай жерде тексеріледі. Циклдың сол уақытында, шарт циклдың денесіне кіру алдында тексеріледі, До-сол ара циклында шарт цикл денесін орындаған соң текеріледі. Бұл сол сияқты болып көрінсе де, До-сол ара циклы цикл денесін орындаған соң шартты тексереді, сондықтан циклдың денесі, тым болмағанда, бір рет орындалады. Егер шарт орындалса, циклдың сол уақыты басынан жалған болып табылады, онда циклдың денесі мүлдем орындалмайды. Төменде С-те екі ілмекті салыстыратын мысал кетіріліген.

осы арада ($x > 0$)

$x = x / 2;$

жасау {

$x = x / 2;$

} Осы арада ($x > 0$);

Осы екі ілмектен жоғарыда, x бастапқыда -1 тең болатын бүтін сан болып саналады деп болжайық. Бірінші циклда, шарт бастапқыдан жалған, және сондықтан циклдың денесі ($x = x / 2$) ешқашан орындалмайды. Осылайша, $x -1$ қалады.

Екінші циклда, циклдың денесі бір рет орындалады. Себебі шарт циклдың денесі орындалған соң сол уақытқа дейін тексерілмейді, сондықтан $x -1$ -ден 0 -ге дейін өзгереді ($-1/2$ шамалары бүтін санға сақтала алынбайды, сондықтан x орнына 0 шамасын алады). Сосын цикл аяқталады, себебі $x 0$ -ден көп емес.

Жаңа тілдер есептеу циклынан итератор циклының пайдасына жиі бас тартады. Бұл итерацияны шамалардың кезеңділігі емес, тізімі бойынша

есептеу ілмегінің нұсқасы. Мысалы, егер сізде 1, 5, 10, 19 тізім бар болса, онда цикл тізімдегі әрбір төрт шама үшін бір рет, төрт рет итерация болады. Python цикл тізімде іріктелетін жердегі мысал болып табылады. Сіз есептеу циклын имитациялау үшін циклдың бұл типін пайдалана аласыз, бұл диапазонды (1, 100) мәлімдеп, 1-ден бастап жоғарғы шекке дейінгі диапазонда іріктейді. C ++және Java бұрыннан цикл үшін есептік нысан ғана болды, бірақ бүгінде екі есептеу және итераторлар ілмегі бар.

Шартты циклдарға қатысты бір соңғы ескертуде циклдың денесі ең кем дегенде, цикл шарттарында пайдаланылатын ауыспалының біреуіне айла-шарғы жасайтын нұсқаулықтар бар. C-ке келесі кодты қарайық.

```
осы арада (x > y)
x = x + 1;
```

Егер x бастапқыда y-ден көп болса, циклдың денесі орындалатын болады. Десе де, y көбірек шамасы кезінде басталатын x салдарынан, x-ке бірізді қоса отырып, ешқашан x-ты y-дан кем немесе тең жасамайды, ал нәтижесі цикл ешқашан тоқталмайтынында тұрады. Бұл шексіз цикл ретінде танымал.

Сізде шексіз цикл бар екеніне көз жеткізу үшін, сіз циклдың денесінде күйде біршама шаманы өзгерте алатын кодтың бар екеніне көз жеткізуіңіз тиіс, сол себепті, ақыр соңында, шарт жалған болуы мүмкін. Келесі C кодты қарап шығайық:

```
осы арада (x > 0)
E ( "% d \ n", x); // Шығару x x = - 1;
```

Бұл цикл ол x 0-ден басталғаннан қалай есептелетіндей көрсетіледі, әрбір шаманы басып шығару. Алайда циклды денесінде екі нұсқаулық бар болғандықтан, біз } { символдарын пайдалана отырып, оларды блоктың ішіне орналастыруымыз тиіс. Өкінішке орай, компилятор бұл кодты талдайды және оператор x = - 1; ілмектен тыс (немесе кейін) бар екенін шешеді. Осылайша, цикл сол арада Пер- шығу Printf өтініш қалыптастырады.

Нәтижесі егер x 0-ден көп болса, Printf бастау үшін, орындайдыжәне шат қайта тексеріледі. Printf x өзгертпеген салдарынан, ол 0-ден әлі де көп және Printf қайта орындалады. Бұл қайта қайталанады, және ешуақытта тоқтамайды (пайдаланушы бағдарламаны үзуге шешім қабылдағанша). Бұл шексіз циклдың себебі логикалық қате болып табылады,бағдарла-

машының логикасы дұрыс болмады. Бұл проблеманы жою үшін, цикл келесі түрде болуы тиіс.

```
сол екі арада (x > y) {  
E ( "% d \ n", x); // Шығару x x = - 1;  
}
```

Шексіз циклдар бағдарламашылар үшін, тіпті үлкен тәжірибесі бар бағдарламашылар үшін жиі проблема болып табылады. Логикалық қатенің бұл типін жиі анықтау қиынға соғады, және бағдарламашы ERS ашуда бағдарламаны тестілеу кезінде қатені тауып, бұл "повесить" болып көрінеді. Бұл жағдайда цикл ешуақытта аяқталмайды, сол себептен бағдарлама келесі қадамға жылжымайды.

Бұрыннан шыққанды еске салайық, Python бастау...соңы және {...} сияқты белгілеменің арнаулы блокын пайдаланбайды. Осының орнына шегіністер пайдаланылады. Сондықтан бұл жағдайда бұрыннан шыққан шексіз цикл, бәлкім, орын алмайды, өйткені шегіністерді шешу керек.

Ішкі бағдарламалар және ішкі бағдарламалардың шақырулары

Ішкі бағдарлама оны шақыра алатын кодтан бөлек кодтың жиынтығы болып табылады. Ішкі бағдарламаны шақыру командасы ішкі бағдарламаны шақырады. Ішкі бағдарлама сіздің бағдарламаңыздың бөлігі болуы мүмкін, немесе бұл басқа бағдарламаның бөлігі немесе ішкі бағдарламалар кітапханасы арқылы қол жетімді кодтың қандай да бір тәуелсіз бөлігі болуы мүмкін. Бағдарламашылардың көбісі бағдарламаларды ішкі бағдарламалар серияларының түрінде жазады. Сөйтіп, бағдарламаны жобалау, жазу және ретке келтіру оңай болады, себебі бағдарламашы сол арада бір ішкі бағдарламаға ғана назар аударуы тиіс. Тұжырымдама модульдің бағдарламалау ретінде белгілі, мұнда модульдер кодтың біршама шағын және тәуелсіз кесектер болып саналады.

Төменде С. В С, тіліне жазылған екі ішкі бағдарламадан қысқаша мысал келтірілген, ішкі бағдарламалар функциялар деп аталады. Бастының функциясы бағдарламада бірініш орындалу болып табылады. Мұнда басты өзінің алғашқы PRINTF (шығу) өтінішін орындайды, сосын Foo функциясын шақырады. Foo функциясы оның меншікі өтінішін шығаратын және тоқтататын болады. Foo аяқталған кезде, бастысы Printf соңғы операторы болып табылатын Foo шақырудан кейін бірден нүктеден қайта жаңартылады. Негізгінің функциясы сосын өз жұмысын тоқтатады және бағдарлама аяқталады.

```
Шынайы емес Foo () {  
E ( "біз в обую Ия?!");
```



```
}
```

```
бос функция( ) {  
Е ( " . Сәлем, әлем. Біз Foo-ды енгізуді бастаймыз."); Foo ();  
Е ( "Енді біз Foo-ден оралдық.");  
}
```

Бұл бағдарламаның шығаруы келесі түрде болады:

Сәлем, әлем. Біз Мы Foo енгізуді бастаймыз. біз Foo? Ия!

Енді біз Foo-ден оралдық.

Бұл бағдарламаны екі функцияда жазу үшін ешқандай себеп жоқ шығар; біз бастыда барлық үш PRINTF өтінішті жазудың орнына. Десе де, бағдарламалардың көбісі бір функцияда жазылу үшін өте күрделі.

Бұрынғы мысалда Foo () функциясын шақыру. Домалақ жақшалар шақыратын функция (негізгі) және шақырылатын функция (Foo) арасындағы байланыс құралы ретінде қызмет етеді. Бұл коммуникация ауыспалылар және олардың шамалары түрінде жеткізіледі. Біз оларды параметрлер деп атаймыз. Мысалы, егер функция шамадан квадрат түбірін есептейтін болса, біз функцияның шамасы біз квадрат түбірді қалайтынымызды айтуымыз тиіс. Егер функция SQRT шақырса, біз 5-тен квадрат түбірін есептеу үшін SQRT (5) немесе x ауыспалыда сақталатын шамадан квадрат түбірін есептеу үшін SQRT (x) сияқты белгілемелерді пайдалана отырып, функцияны шақыра алушы едік. Томенде Сте функцияда параметрлер берілісінің мысалы келтірілген. Функция шаманы есептейді (кері параметрдің) және нәтижесін шығарады.

Күшін жою Reciprocal ретінде айқындалады (флоат x) {егер (x == 0)

Е ("Онда Там өзара 0 жоқ"); тағы Е ("Өзара болып табылады % F", 1,0 / x);

```
}
```

Мұнда функция құбылмалы нүктесі бар шаманы береді. Функция әрбір жеке параметрі үшін басқа хабарламаны шығаратын болады. Мысалы, егер $x = 1,0$, бұл шығу нені жасайды? Егер $x = 10,0$ болса, не істеу керек? Егер $x = 0$ болса не істеу керек?

Арнаулы мақсаттағы нұсқаулықтар

Оларға операциялық жүйенің шақырулары, жолдармен операциялар, процедураның компьютерлік графикасы, операциялардың және қателіктерді өңдеу процедураларының кездейсоқ саны.

Жоғарыда орындалатын пәрмендер тізімінен басқа кейбір бағдарламалау тілдерінде сондай-ақ *ations* айқын айнымалысын талап етеді, Pascal, C және Java секілді тілдерде айырмашылықтар

```
Var x, y : integer;  
Var z : real;  
  
int x, y;  
float z;  
  
var x, y, z;
```

14.6 сурет. Айнымалыларды хабарлаудың үш формалары.

теруді талап етеді, яғни, бағдарлама барысында қалатын нақты мәндер түрі хабарланады. JavaScript-те айнымалылар аты бойынша хабарлануы тиіс. 14.6-суретте x, y және z үш айнымалылар үш тілде хабарланады: Pascal (жоғарыда), C (ортасында) және JavaScript (төменде). В C, INT Паскальдің бүтін шынайысын (он нүктелі сан) білдіреді.

Кейбір тілдер өзіндік түрлерді анықтауға мүмкіндік береді. Ada-да сіз басқа түрдің кіші жиынтығы болуы мүмкін түрді хабарлай аласыз. Мысалы, біз 0..100 рұқсат етілетін ауқымының бүтін сандар кіші жиынтығы үшін түрін анықтауымызға болады. Осылайша, түрдің айнымалысы – бұл кез келген бүтін сан, мұнда сынып түрінің айнымалысы 0-ден 100-ге дейінгі ауқым болып табылады. Java-да түрлер сыныптар деп аталады.

Скриптік тіл

Жүйелік әкімші ретінде бағдарламалау тілінің маңыздылығын түсінесіз, себебі, сіз сценарийлерді жиі жазатын боласыз. Сценарий интерпретацияланатын бағдарлама болып табылады. Сценарийлер көбінесе қысқа келеді (қосымшалармен немесе жүйелік бағдарламалық қамтамасыз етумен салыстырғанда). Бірақ, ірі масштабты қосымшаларға арналған кодты жазу қажеттілігінің болмауына қарамастан, жүйелік әкімші бағдарламалау тілінің пәрмендерін де түсінуі тиіс.

Нұсқаулар түрлері

Қазіргі кездегі сценарийлердің танымал тілдері сценарийлер тілдерінің қабықтарын (мысалы, Bash script - ИНГ тілі және DOS), сондай-ақ, Python, Ruby, Perl және PHP секілді неғұрлым толық әрі күрделі бағдарламалау тілдерін қамтиды. Мұнда біз Bash Shell және DOS тілдеріне

ғана тоқталамыз. Біз көптеген мысалдарды көру үшін нұсқаулардың түрлеріне сипатталынған тілдің ерекшеліктерін арнайы қарастырамыз. Себебі, бұл мәтінді бағдарламалау емес, бұл қималар қабық сценарийін енгізу үшін ғана қажет.

Қабық сценарийіне енгізе алатын кейбір кодтар командалық жолда қолданатын пәрмендер (командалар) секілді. Linux-те, мысалы, сценарий компакт-дискті, LS, гт, эхо және т.б. қамтуы мүмкін. Дәл осылай келесі екі бөлімшелерде көретін пәрмендерді де командалық жолға енгізуге болады. Дегенмен, бірқатар айырмашылықтары бар. Сценарий параметрлер алуы мүмкін, бұл сценарийді неғұрлым мықты етеді.

Тілдің Bash қабықтары

Қабықтың скриптері мәтіндік файлдарда сақталынады. Бұл файлдар оқу мен орындау үшін де болуы тиіс. Осындай файлдарды 745 немесе 755 ажыратымдылықпен орнатқан жақсы. Foo атауы бар сценарийді орындау үшін сіз ./ Foo енгізуіңіз қажет. Бұл сіздің жұмыс каталогыңыздың Foo файлын сақтайтындығын білдіреді. Bash қабықтарының барлық сценарийлері Bash интерпретаторы орындауды ескертетін келесі жолдан басталуы тиіс: #! / Bin / Bash.

Bash сценарийлеріндегі айнымалылар командалық жолдағыдай өңделеді. Сіз VAR = VALUE түріндегі иелендіру операторын қолдана отырып, айнымалыға мән бере аласыз. Оң жақтағы жолдағы * мәні жол болуы тиіс, дәлме-дәл саны немесе мәні негіздегі бүтін санның арифметикалық өрнегі ретінде есептелінеді. Егер жолда бос орындар болса, барлық жолды "" белгілеріне қамтыңыз. Атының алдында \$ айнымалысы бар айнымалының мәніне қолжетімділік алу үшін, біз төмендегі арифметикалық өрнектердің бүтін сандық негіздеріне қараймыз. Төменде бірнеше мысалдар келтірілген.

X = 0

Y = "Сонда сәлем" NAME = Frank Z = \$ X

MESSAGE = "Y \$ NAME"

X жолы 0 мәнін қамтиды, Y жолы "Сонда сәлем" және NAME жолы "Франк" қамтиды, тіпті тырнақшалар иелендіру операторында қолданбаған болса да. Z мәнді алады.

* Айнымалылардың аттары бас әріптермен жазылмау қажеттілігіне назар аударыңыз. Мен нұсқаудан айнымалыны оңай ажырату үшін бас әріпті қолданатын боламын.

Қазіргі кезде X-де 0 сақталады. ЖОЛДАУ Y-ге сақталынатын мәндер жолын (Сонда сәлем), алады, получает строку значения, хранящегося в Y (Привет там), NAME-де сақталынатын бос орын мен мәнді (Frank) немесе "Frank сонда сәлем" алады.

Негізде бүтін операциясы қиын, себебі, егер ол айнымалы мәндерге ие болса, ол өзіндік синтаксиске қамтылуы тиіс: \$ (()). Мысалы, $Z = X + 5$ болуын орнату үшін, сіз $Z = \$ ((X + 5))$ қолдануыңыз қажет, сонымен қатар, X-ті ұлғайту үшін сіз $X = \$ ((X + 1))$ қолдануыңыз қажет. Арифметикалық өрнектерге арналған заңды операторлар +, -, *, / және % болып табылады (% модуль бойынша операцияларды орындайды, яғни, ол бөлгеннен қалдықты қамтамасыз етеді). Бөлу операциясы (/) тек бүтін сандармен ғана жұмыс істейді және жеке бүтінді қайтарады. Мысалы, егер $X = 5$ және $Y = 2$ болса, онда $Z = \$ ((X / Y))$ Z-ті 2 жоққа орнатылуына әкеледі,

2.5. $Z = \$ ((X \% Y))$ операциясы 1-ге орнатылған Z-ке әкеледі, себебі, $5/2$ 2 1 қалдығымен.

Эхо шығыс тұжырымы. Сіз мұны оқулықтан көргенсіз. Эхо оператор шығаруға арналған элементтер тізімін қамтиды. Бұл тізім айнымалыларды, әріптік мәндерді және Linux пәрмендерін қамтуы мүмкін. Егер сіз бұл тізімді "-ге орналастырсаңыз, шығу дәлме-дәл мәнінде жүзеге асырылады (яғни, айнымалылар мәндері қайтарылмайды, олардың орнына сіз айнымалылардың аттарын аласыз). Мысалы, егер NAME = Франк және жасы = 53 болса, онда

```
$ NAME Эхо $ AGE құрайды, Франк 53 шығуын қамтамасыз етеді  
бірақ нұсқау  
'$ NAME Эхо $ AGE' құрайды  
шығуын қамтамасыз етеді  
$ NAME $ AGE құрайды.
```

Сіз сондай-ақ тырнақшасы жоқ дәл сондай нәтижені қамтамасыз ететін "" шегіндегі өзінің шығуын орналастыра аласыз. Көптеген жағдайларда, "" эхо өтінішке түсірілуі мүмкін. Дегенмен, келесі екі эхо-өтініштерді қарастырамыз:

```
$ NAME Эхо $ AGE құрайды "$ NAME $ AGE" құрайды
```

Бірінші эхо тұжырым үш аргументті (параметрлерді) қамтиды, ал екінші эхо тұжырым тек біреуін қамтиды. Сонымен қатар, "" қолдану Bash 'разлет' файлының атауын басып тастайды. Мысалы, * эхо Bash бірінші кеңейтуге, * барлық элементтері жергілікті каталогта болуға әкеледі, сосын эхо оларды шығарады, сондықтан * эхо Ls немесе Ls * секілді. Алайда "*" эхо дәлме-дәл мәнінде шығу *, файлдың атын кеңейту орындалмайды.

Сіз эхо-ны Linux басқа пәрменін шығара отырып, дәлме-дәл хабарламаны шығаруды бірікту үшін қолдана аласыз. Мысалы, егер сіз ағымдағы күн / уақытымен бірге "күн және уақыт болып табылады" шығарғыңыз келсе, сіз эхо өтініш пен операция күнін біріктіре аласыз. Дегенмен, дәлме-дәл мәнінде "күні" өзін шығарудан эхо-ны сақтау үшін сіз Bash-ты күн пәрменін орындауы тиіс деп көрсете аласыз. Бұған кері тырнақшаның көмегімен қол жеткізіледі. Мысалы, оператор

эхо «күн»

дәлме-дәл шығу "күн". Дегенмен, өтініш

эхо күн `date`

Linux-ке күн пәрменін орындауды, сосын эхо мәнге өзінің шығуын қоюды мәжбүрлейді. Нәтижесі келесідей болады:

"күн Бс 29 наурыз 2012 10:35:32 EDT".

Енгізу оқу жауабы арқылы жүзеге асырылады. Бұл тұжырым әрқашан болуы керек, әрі қарай айнымалының аты жүреді. Мысалы, "NAME оқу" пайдаланушыдан енгізуді алу және енгізілген мәнді NAME айнымалысына орналастыру үшін қолданылатын болады. Кірме мәні бар сценарийді оператор орындағанда, жеке деректерді бос жолдағы жыпылықтайтын курсорға әкеледі. Пайдаланушы ненің болып жатқанын білмеуі мүмкін болғандықтан немесе бағдарлама күтетіндіктен, кез келгеннің алдында ұсынатын өтініші бар өтініштің болуы маңызды. Контексті бұл пайдаланушыға жасауына рұқсат беретін шығару операторы. Келесі мысал пайдаланушыдан олардың атын, мәнді енгізуді, сосын персоналды шығу өтінішін қамтамасыз етуді сұратады. Кірме өтінішіндегі айнымалының аты \$ алдында келмейтіндігіне назар аударыңыз, ал \$ шығу хабарламасында қолданылады.

эхо Оқу атын енгізіңіз NAME

эхо Сәлем \$ NAME, сізбен кездескеніме қуаныштымын!

Пайдаланушы оқу операторына бірдеңені енгізуі мүмкін. Мысалы, олар өзінің атын немесе басқа жолды толығымен немесе бірнеше жолды енгізуі мүмкін. Жоғарыдағы кодтағы ешнәрсе бұған кедергі келтірмейді. Егер кіру онда орын алса, жол тырнақшаға алынуын талап ететін NAME = "Frank Zappa" секілді иелендіру операторына қарағанда, кодта ешқандай арнайы механизм қажет етілмейді. Егер пайдаланушы жоғарыда аталған код үшін Фрэнк Заппаны енгізетін болса, онда шығуы біз күткендей болады, "Сәлем Фрэнк Заппа, сізбен кездескеніме қуаныштымын!".

Қазіргі сәтте біз көргендеріміздің барлығын біріктіреміз. Келесі сценарий төрт мәннен тұратын орташа мәнді есептейтін болады. Екі иелендіру операторда жақшалардың қолданылуына назар аударыңыз. Есіңізге салайық, айнымалыларды қамтитын арифметикалық операцияны орындау үшін, \$ ((...)) белгілеуін қолдануға болады. Дегенмен, осы жағдайда ажыратуды орындауды немесе операциялар модасының алдында num1, num2, num3 және Num4 теңегіміз келетіндіктен, біз жинақтауды домалақ жақшаның тағы бір қабатына орналастырамыз. Жақшаларсыз бөлу (Num4 / 4) қосымшалар алдында орындалады.

#!/ Bin / Bash

эхо Өтініміз, сіздің бірінші оқу нөмірін енгізіңіз num1

эхо Өтініміз, сіздің екінші оқу нөмірін енгізіңіз num2

эхо Өтініміз, сіздің үшінші оқу нөмірін енгізіңіз num3

эхо Өтініміз, сіздің төртінші оқу нөмірін енгізіңіз Num4

фактор = \$ (((num1 + пит2 + num3 + Num4) / 4)) қалдық = \$ (((num1 + пит2 + num3 + Num4)% 4))

эхо-сигналды орташада есеппен / 4 қалдығының \$ қалдығы бар \$ факторы құрайды.

Bash-тағы шартты оператор өте ерекше синтаксиске ие. Осыған байланысты, ол іске қосу сценарийлерін дұрыс алу үшін енгізу скриптерлеріне арналған шақыруды жиі тастайды. Синтаксисті ұстанатындығыңызға көз жеткізіңіз. Өтініш формасы ", егер [жағдай]; онда әрекет (s); фи " болып табылады. Жартылай өткізгіш бағаналар осында көрсетілген жерлерде орналастырылуы тиіс. Сонымен қатар, егер өтініш суретте көрсетілгендей болса, әр түрлі компоненттер арасында аралықтар болуы тиіс. Ақырында, егер өтініш болса, онда "фи" сөзі аяқтауы тиіс. Әрекеттер үтірлі нүктемен бөлінген жекелеген тұжырымдар болып табылады. Мысалы, сіз иелендіру операторларына, кірме операторларына, шығару

операторларына және тіпті енгізілген-өтініштерге де ие бола аласыз.

Шартты оператор шартты талап етеді. Шарттың кең таралған екі формасы бар:

- салыстыру айнымалысының мәні
- файлдың аты filetest

Бірінші жағдайда, айнымалының аты \$ алдында келеді. Салыстырулар жолдарды салыстырулар үшін == немесе != біреуі немесе сандық салыстырулар үшін -eq, -ne, -gt, -lt, -ge, -le біреуі болып табылады (тең, тең емес, артық, кем, артық немесе тең, кем немесе кем). Мән дәлме-дәл мән, басқа айнымалы (аты \$ алдында келетін) немесе арифметикалық өрнек бола алады.

Екінші жағдай файлдың аты бар filetest шарты болып табылады. Filetest - d (пункт каталог болып табылады), - e (файл бар), - f (файл бар және қарапайым файл болып табылады), -h (пункт символикалық сілтеме болып табылады), -r (файл бар және оқылатын болып табылады), -w (файл бар және жазу үшін қолжетімді) және -x (файл бар және орындалатын болып табылады) біреуі болып табылады. Filetest шарт жокқа шығарылуы үшін ! таңбасының алдында болуы мүмкін. Яғни, foo.txt оқылатын болса, тест - r foo.txt сұратады, ал егер foo.txt оқылмаса, ! -r foo.txt сұратады. Кейбір мысалдар шарттарды орындайды. Шарттың әрбір бөлігінің арасындағы бос орындарға назар аударыңыз. Осы бос орындарсыз, сіздің сценарийіңіз катені түрлендіретін болады және орындамайды.

```
[$ NAME == "Frank"] [$ X -gt Y $]
```

```
[$ X -eq 0]
```

```
[$ STUDENT1! = $ STUDENT2] [-x $ FILENAME]
```

```
[$ ! -r $ ФАЙЛ2]
```

Толық мысал төменде келтірілген. Бұл жағдайда өтініш, \$ NAME айнымалы "Франк" мәнімен салыстырып қаралады және егер олар тең болса, онда нұсқау "Сәлем Франк" хабарламасын шығарады және X айнымалысына 1-ді қосады. ФИ соңғы жағдайларда if-оператормен аяқталады.

```
егер [$ NAME == "Frank"]; онда echo Hi Frank; $ X = $ ((X + 1)); fi
```

Кейбір жағдайларда бұл жағдай логикалық немесе OR-мен үйлесімде бірден көп салыстыруды талап етеді. Мысалы, егер айнымалының мәні екі басқа мәндер арасында жататын болса, тексеру және көру үшін, сіз айнымалыны екі мәндерге қарсы тексеруіңіз қажет. Егер мән X айнымалысында сақталса және 0-ден 100-ге дейінгі ауқымда болғанда біз көргіміз келсе, $\$ X -ge 0$ және $\$ X -le 100$ сынара едік. Синтаксис бұл жағдайда неғұрлым тән болады. Мұндай шарт шартты компаунд деп аталады және барлық жағдайдың айналасында `[]` қосымша теруді талап етеді. ЖӘНЕ мен NEMECE үшін қолданылатын таңбалар AND (ЖӘНЕ) үшін C мен Java, `&&` және OR (NEMECE) үшін `||` қолданылатын таңбалар. 0-ден 100-ге дейін X-ті тексеру үшін, біз келесіні қолданатын боламыз: `[[\$ X -ge 0 && \$ X -le 100]]`.

Спецификатор есепке `if`-ті қоса алады және осы жағдайда "тағы" сөзі тағы пункттегі әрекет(тер) ілесетін сол пункттегі әрекет(тер)ден кейін пайда болады. "Фи" басқа пунктен кейін пайда болады. Төменде үш мысал келтірілген. Екінші мысалдың негізінде ешқашан мәннің жоқтығына назар аударыңыз, себебі, Linux-те есептеудің құбылмалы нүктесі бар ештеңе жоқ, сондықтан, $\$ ((1 / X))$ операциясы шынайы керіні қамтамыз етпейді.

егер $[\$ X \$ -gt Y]$; онда $Y = 0$; тағы $X = 0$;

Fi

егер $[\$ X -ne 0]$; онда $Y = \$ ((1 / X))$; эхо Кері $\$ X \$ Y$ құрайды; әйтпесе эхо кері 0-ді есептеп шығара алмайды;

Fi

егер $[\$ NAME == "Frank"]$; сосын эхо "Hi Frank"; $X = \$ ((X + 1))$; тағы эхо Kim ?; $X = 0$;

Fi

Сіздің пункті немесе басқа фи өтінішті жекелеген жолдарға орналастыра алмайтындығыңызға назар аударыңыз, олар оқуды қолайлы ету үшін осылай жазылған.

Сондай-ақ, енгізілген, егер-онда-басқа өтініші бар, мұнда "тағы, егер" Elif секілді жазылады. Құрылымы келесідей көрінеді:

егер [жағдай]; онда

әрекет (s); Элиф [жағдай], онда

әрекет (s); Элиф [жағдай], онда

... тағы

Fi

әрекет (s);

әрекет (s);

Егер 14.5 суретті қарайтын болсақ, Паскаль тілінде басқа өтініш бар. Bash-тағы балама тұжырым 14.7 суретте көрсетілген.

Bash тіліндегі сценарийлерде қолжетімді циклдардың екі түрі бар. Цикл үшін тізімді қайта таңдайтын итератор циклы болып табылады. Тізім бірнеше заттардың біреуінен тұруы мүмкін. Біріншіден, ол мәндердің нөмірленген тізімі болуы мүмкін. екіншіден, бұл параметрлер тізімі түрінде скриптке қойылған тізім болуы мүмкін (бұл осы ыбөлімде талқыланады). Үшіншіден, бұл қойылымды түрлендіретін, яғни файл атының кеңеюін түрлендіретін тізім болуы мүмкін.

Петляға арналған Bash негізгі формасы:

тізімдегі айнымалы үшін; өтініш(тер) жасау; жасалған

Мұнда айнымалы цикл индексі болып табылады. Цикл индексі тізімдегі әрбір мән бойынша, бір қабылдағанда біреуден қабылдайды. Цикл индексі циклдың сіздің денеңіздегі пікірлерге сілтеме жасауға болады. Мысалы, себебі мен 1 2 3 4 5-те; \$ I эхо жасау; жасалған цикл итерациясында бір-бірден жекелеген жолдарда 1, 2, 3, 4 және 5-тен әрбіреуі шығарылатын болады.

```
if [ $grade -ge 90 ];  
    then letter=A;  
elif [ $grade -ge 80 ];  
    then letter=B;  
elif [ $grade -ge 70 ];  
    then letter=C;  
elif [ $grade -ge 60 ];  
    then letter=D;  
else letter=F;  
fi
```

14.7 сурет. Bash-тағы енгізілген егер-онда-басқа.

Негізінен цикл екі мақсатта қолданылады. Біріншіден, әмбебаптандыруды қолдана отырып, директорияда элементтерді іріктеу. Екіншіден, параметрлерді өңдеу үшін оларды іріктеу. Параметрлер туралы осы бөлімде кейінірек қарастыратын боламыз. Келесі цикл глобирлеуді қолданудың мысалы болып табылады. Бұл жағдайда цикл ағымдағы каталогтағы әрбір пункт бойынша ұзын тізімді орындайды. Мұны Ls -l * қолдана отырып, оңай жүзеге асыруға болады.

```
for file in *.*; do ls -l $file; done
```

Циклдағы глобирлеу оператор `filetest` шарттарының бірін қолдана отырып, файлды тестілейтін цикл деенсіне шартты операторды орналастырған кезде анағұрлым пайдалы болады. Әрі қарай, мысалы, каталогтағы әрбір мәтіндік файл орындалатын болып табылатындарды тексеріп шығарады.

```
for file in *.txt; do
if [ -x $file ];
then echo $file;
fi
done
```

Тереңдету қажетті болып табылмайды, бірақ оқылымдылықты қосады. Бұл цикл ағымдағы каталогқа барлық `.txt` файлдарды іріктейді және орындалатын файлдардың атын бейнелеп көрсетеді.

Циклдың басқа түрі – бұл шартты цикл. Бұл цикл `C` петлясына балама. `while [ condition ]; do statement(s); done`

Циклдың шексіз болу мүмкіндігін есіңізге саламыз. Сондықтан, цикл денесінің белгілі бір сәтте жағдай шынайыдан жалғанға өзгеретін кодты қамтуы өте маңызды. Цикл итерациясының санын басқаратын пайдаланушымен өзара қарым-қатынасты қамтамасыз ету үшін қолданудың бір дұрыс циклы. Мысалы, біз файлдар аттарының тізімінде бірқатар үдерісті орындай аламыз. Біз пайдаланушыдан файл атын енгізуді сұрай аламыз. Циклдың жалғасу-жалғаспауын бақылау үшін біз пайдаланушыны шығу үшін арнайы кілт сөзді енгізуді сұрай аламыз, мысалы, "шылым шегуді тастау". Бұл жағдайда "шылым шегуді тастау" қадағалаушының мәні деп аталатын болады. Келесі сценарий пайдаланушыдан файлдардың аттарын бірнеше рет сұратады, оқу мен жазуға арналған файлдарды шығарады, пайдаланушы "шылым шегуді тастау" енгізген кезде циклдан шығарады. Егер өтініш өтінішті бір рет санаса, шарт кодтың соңында жиынтық шығу үшін шынайы.

```
#!/bin/bash
```

```
COUNT = 0
```

```
echo Файл атын енгізіңіз, шығу, FILENAME оқу шығу үшін
сол кезде болғанда [$ FILENAME! = Шылым шегуді тастау]; жасау
егер [[-r $ FileName && -w $ FILENAME]];
сосын -l $ сол шкала FILENAME; COUNT = $ ((COUNT + 1));
жасалған
```

Fi

echo Файл атын енгізіңіз, шығу, FILENAME оқу шығу үшін

echo Оқу мен жазуға арналған \$ COUNT файлдары болды

алдыңғы мысалда екі пәрменнің циклдың алдында және циклдың соңында пайда болғандығына назар аударыңыз. Неліктен эхо мен өтініштерді оқу екі жерде де пайда болу керек деп ойлайсыз? Егер де сіз екі өтінішті циклдың алдына түсіргенде не болатындығын ойлаңыз. Циклдың шартына қол жеткенде, \$ FILENAME ешқандай мәнді қайтармайды, себебі, цикл ешқашан да кірмеді. Осылайша, біз файл атының бастапқы мәнін қамтамасыз етуіміз қажет. Егер біз эхо мен оқуды циклдың ішіне түсірсек, онда сіз бастапқыда енгізген файлдың кез келген аты FILENAME айнымалысында қалады. Осылайша, петля шарты әрқашан шынайы болып табылады және шексіз циклға иеміз!

Сіз, әдеттегідей, қалаған цикл уақытын қолдана аласыз. Пайдаланушы шығатын шартты көрсететін (мысалы, шылым шегуді тастау) мәнді енгізгенін күткенде немесе сіз шекке жеткен кезде тоқтату қажет есептеулер сериясын жасағыңыз келгенде пайдаланушы енгізетін бірқатар деректер бойынша міндетті орындау. Бірінші жағдай алдыңғы кодта көрсетілген. Келесі мысалда 1000-нан 2 есе аз өкілеттілікті есептеп шығарады немесе 1000-нан көп немесе тең бірінші табылғанды шығарады.

```
#!/bin/bash
```

```
VALUE=1
```

```
сол кезде болғанда [$ VALUE -lt 1000]; жасау
```

```
VALUE = $ ((VALUE * 2));
```

```
жасалған
```

```
echo 1000-нен көп екі бірінші дәреже $ VALUE құрайды
```

Bash скрипттер тапсырылған параметрлер болуы мүмкін. Олар командалық жолға сценарий атынан кейін енгізіледі. Сценарий ./script <Enter> белгілеуін қолдана отырып орындалады. Параметрлер ./script 5 10 15 <Enter>-дегідей <кіру>-ге дейін қосылуы мүмкін. Бұл параметрлер сценарийлерде қолданылуы мүмкін, егер олар айнымалылар болса, пайдаланушы берілген мәндермен инициализацияланады. Тапсырма параметрлерінің үш әр түрлі тәсілі бар. Біріншіден, \$ # шарт жағдайында параметрлер санын сақтайды. Бұл пайдаланушының қажетті параметрлер санын бергендігін анықтау үшін егер өтініште қолданылуы мүмкін. Екіншіден, әрбір жеке параметр \$ 1, \$ 2, \$ 3 және әрі қарай қолдана оты-

рып қолжетімді. Егер сіз екі параметрді күтсеңіз, сіз \$ 1 және \$ 2 секілді оларға қолжетімділік алар едіңіз. Соңында, \$ @ параметрлердің барлық тізімін қайтарады. Сіз циклда \$ @-ды тізім түрінде қолдануыңыз қажет. Келесі скрипт екі параметрлердің үлкенін шығарады. Егер пайдаланушы екі параметрді дәл көрсетпесе, онда қате туралы хабарлама алады.

```
#!/bin/bash
егер [$ # -ne 2];
онда осы сценарийдің echo екі параметрді талап етеді; elif [$ 1 -gt $ 2];
онда echo Сіздің бірінші параметріңіз көбірек; тағы echo Сіздің екінші
параметріңіз көбірек;
Fi
```

Келесі сценарий файлдар атының тізімін күтетін болады. Егер осындай тізім қарастырылмаса, яғни, \$ # 0-ге тең болады, егер ол қате туралы хабарлама шығарса. Кері жағдайда ол тізімді іріктейді және орындалатын мәртебеге арналған әрбір файлды тексереді және орындалатын болып табылатын барлық файлдардың ұзын тізімін қамтамасыз етеді. Шарты тағы циклға ие және циклында өтініш бар бұл сценарийдің басқа өтініші бар екендігіне назар аударыңыз. Сценарийде екі өтініштің болғандығына байланысты екі Fi тұжырымына қажеттілік бар.

```
#!/bin/bash
егер [$ # -eq 0]; онда
echo емес Көрсетілген параметрлер жоқ, жалғастырыла алмайды;
тағы
$ @-дағы файл үшін; жасау
егер [-x $ FILE]; сосын -l $ файлдың сол шкаласы;
```

```
Fi
жасалған
Fi
```

Параметрлер ретінде берілген нөмірлер тізімін күтетін сценарийді аяқтайық. Сценарий есептеп шығарады және тізімнен орташаны шығарады. Скрипт ОРТАША деп аталады деп болжай отырып, сіз ./avg 381 56 10 18 266 531 тере отырып, командалық жолдан оны шақыра аласыз.

```
#!/bin/bash
егер [$ # -eq 0]; сосын параметрлер эхо, орташа мәнді есептеп шығара
```

```

алмайды; тағы
SUM = 0;
NUMBER $ @-дегі ; жасауSum = $ ((SUM + нөмір));
жасады AVERAGE = $ ((SUM / $ #));
echo $ #-дегі орташа мән $ AVERAGE құрайды;
Fi

```

Осы мысалда барлық енгізілген сандар пайдаланушы сценарийді орындаған кезде көрсетілгендігіне назар аударыңыз. Егер біз пернетақтадан мәндерді енгізгіміз келсе не істеу керек (оқу пәрменінің көмегімен)? Ол қалай ерекшеленеді? Бұл мәселе шолу бөліміндегі жаттығуда қалады. Дегенмен, міне бірнеше сыбыр сөз. Сіз циклдың сол уақытын қолдануыңыз керек. Енгізілген барлық сандарды оң деп санайық. Циклдың шарты [$\$ \text{CAN} - \text{gt } 0$] болады, пайдаланушы 0 немесе теріс санды енгізген кезде, цикл токтатылады. Сіз эхо өтінішті қосуыңыз (пайдаланушыдан сұрату) қажет және бірінші кіру және әрбір келесі кіру үшін қолжетімділік алу үшін циклға дейін және кезінде өтінішін оқыңыз. Bash-тағы қабық сценарийлерінің соңғы бір комментарийі. Жалпы, егер барлық сіздің нұсқауларыңыз жеке жолдарға орналастырылған болса қажет емес. Бұл "сонда", "жасау" және "басқа" секілді сөздерді жекелеген сызықтарға орналастыруды талап етеді. Жоғарыда келтірілген сценарийді келесідей түрде де қайта жазуға болады:

```

#!/bin/bash
егер [ $\$ \# -\text{eq } 0$ ], сосын
тағы
echo емес, Ешқандай параметрлер орташаны есептеп шығара алмайды
SUM = 0
$ @-дағы алмастырылатын нөмір үшін
жасау
SUM = $ ((SUM + NUMBER))
жасалған
Орташа = $ ((SUM / $ #))
echo $ #-дегі орташа мән $ AVERAGE Fi құрайды

```

Қабықтың енгізу Scripter ретінде, бірақ бұл үтірлі нүктені қосу үшін әрқашан қауіпсіз.

MS-DOS

Бұл бөлімде біз MS-DOS-қа арналған арнайы пәрмендер мен синтак-

систі қарастырамыз. Сіз осы бөлімді оқу үшін Bash тілінің қабығын оқып шығуыңыз керек, петля секілді түсініктер мұндай қайталанбайды. Оның орнына сіз оны біледі деп болжанады, сондықтан біз пәрмендер, синтаксис және мысалдармен ғана шектелеміз.

Bash тіліндегі секілді DOS жолдар мен бүтін сандарды сақтай алатын айнымалыларға мүмкіндік береді. Айнымалылардың үш түрі бар: скрипт орындалып жатқанда пәрмендер (командалар) сызығынан берілетін параметрлер, қоршау айнымалылар (операциялық жүйе немесе пайдаланушы анықтайды), айнымалылар циклына арналған индекс. Параметрлер % N көмегімен белгіленеді, мұнда n параметр санын білдіреді. Яғни, % 1 бірінші параметр, 2% екінші, % 9-ге дейін тоғызыншы параметр үшін. Сіз тоғыз параметрмен шектелмегіңізге қарамастан, көбірек параметрлерді қолдану қиын (% 10,% 11 және әрі қарай сілтемелер секілді оңай емес). Қоршау айнымалылар %аты% нотацияларымен пайда болады (сандық мәндерден басқа, мысалдарды қараңыз). Петля үшін айнымалылар %нотация атымен пайда болады. Айнымалыға мән тағайындау үшін, иелендіру операторы " x жиыны = сәлем" секілді сөздер жиынын қолданады. Басқа айнымалыда сақталынған қосымша айнымалыға мән беру үшін "X жиыны =% Y%" белгілеуін қолданыңыз. Айнымалы атының жанындағы бұл пайыз белгілері айнымалыға құнды интерпретациялайды; кері жағдайда, интерпретатор атты дәлме-дәл қолданады, мысалы, "x жиыны = y" у айнымалысында сақталынған мәнді емес, "y" мәнін сақтау үшін x орнатылатын болады. Дегенмен, бұл келесі азат жолда сипатталынғандай арифметикалық мәндермен жұмыс істейтіндігіміз арқылы ерекшеленеді.

Айнымалылардағы мәндер әдепкі қалпы бойынша мәндерде сандар болса да жолдар секілді өңделеді.

Келесі екі пәрменді қарастырайық.

a = 5

Б жиынтығы =% в% + 1

Б айнымалысындағы 6 мәнді сақтауға ұқсас. Нұсқаулар 5 + 1 жолын б-да сақтау. Бұл теңдік белгісінің оң жағындағы элементтер әдепкі қалпы бойынша жолдар ретінде қарастырылады, сондықтан % A% + 1 белгілеуі "+1" жолда сақталынатын мәндерді біріктіру жолымен жол операциясы саналады. Әдепкі қалпы бойынша осы мінез-құлықты өзгерту үшін сіз жолдар біріктіруін емес, арифметикалық операцияда өрнекті қарау үшін жиынтықта / а көрсетуіңіз қажет. Осылайша, екінші тұжырым б-ға дейін б-ны дұрыс орнататын "орнату / АВ =% в% + 1" болады. Арифметикалық тұжырымдар үшін әр түрлі айнымалылардан пайыз белгілерін

түсіретіндей, "орнату / $AB = a + 1$ " де солай жасайды.

Input екі кішігірім вариациялары бар теру пәрменін де қолданады. Біріншіден, / P параметрі оператор жиынтықты пайдаланушыға ұсыну үшін қолданылады. Екіншіден, теңдік белгісінің оң жағының мәні немесе өрнегі болмайды, бірақ оның орнына кіру алдында шығарылатын хабарламаны тудыратын болады. Егер оң пәрменді түсіретін болса, пайдаланушы тек жыпылықтайтын курсорды көреді. Келесі мысалда енгізуді қалай қолдану керектігі көрсетілген.

```
set /p first=Enter your first name
set /p middle=Enter your middle initial
set /p last=Enter your last name
set full=%first% %middle% %last%
```

Өтініштің соңғы жиынтығындағы әрбір айнымалылар арасында бос орындар үш жолды біріктіру Frank Zappa V-дағы секілді олардың арасындағы кеңістік қойылатынына назар аударыңыз.

DOS-тағы шығу өтініштер Linux секілді эхо-пәрменді қолданады. Шығу өтініштері әріптік мәндер мен айнымалыларды үйлестіретін болады. Ажырата білу үшін айнымалыда сақталатын Hello немесе мән секілді әріптік мәнді пайыз белгісінің ішіндегі айнымалы атына орналастыру қажет. Егер айнымалының аты Фрэнк Заппаны сақтайды деп болжайтын болсақ, "Эхо сәлем аты" өтініші "сәлем аты" дәлме-дәл шығарады, ал "эхо сәлем% аты%" "сәлем Frank Zappa" шығаратын болады.

DOS сценарийі әрбір орындалатын операторды шығару үшін бапталған. "осы функцияны ажырату" үшін сіз @ECHO аулақ қосуыңыз қажет. Экранда емес, файлға шығаруды жөнелту үшін файлға жазу үшін эхо-өтінішке "> файл аты" және файлға қосу үшін ">>" файл атын қосыңыз. Мысалы, егер Фрэнк Заппаның дүкенінің тағы да аты деп болжайтын болсақ. Үш келесі шығу өтініштері экранға, greetings.txt файлға сәлемдесу хабарламасын қалай шығаратындығын және greetings.txt файлға қосатындығын көрсетеді.

```
echo сәлем% аты%
echo сәлем аты%%> greetings.txt эхо сәлем% Аты% >> greetings.txt
```

Шартты оператор шарттың үш түріне жол береді. Сіз соңғы орындалған бағдарламада көрсетілген көрсеткіштер ретінде қате деңгейін тексере аласыз, / мәнінің екі айнымалысын салыстырыңыз және егер файл болса, тексеріңіз. Шарттардың осы түрлерінің әрбіреуі үшін сіз жоқ сөзімен

шартты алдына қоя аласыз. Салыстыру сандық салыстырмалылар үшін фас (=), NEQ (тең емес), LSS (<), Leq (<=), ГТП (>) және GEQ (> =) және жолдар үшін == болып табылады. Шартты оператор міндетті емес басқа орынға ие. Барлық егер-өтініш (соның ішінде басқа қосымша пункт) сценарийдің бір жолында пайда болуы тиіс, осылайша, ол қандай да бір пункт орындай алуы мүмкін әрекеттер санын шектейді. Егер сіз қосымша басқаны қолдансаңыз, екі сосын пункті және () ішінде пунктті орналастырыңыз. "сосын" сөзі С-дағыдай түсірілген. if-өтініштердің бірқатар мысалдары.

Егер эхо қате коды 0 болса, келесі бағдарлама дұрыс орындалады.

Егер% x% Гтп 5 (эхо X 5-тен көп) қалғаны (эхо X 5-тен көп емес) Егер% x% LSS 0 жиынтық / а x = 0-% x%

Егер жоқ% аты% == Франк (эхо, мен білмеймін, сіз)

С болғанда: \ Users \ foxr \ foo.txt дель C: \ Users \ foxr \ foo.txt

DOS циклды қамтиды, бірақ циклдың жұмысы кезінде қамтымайды. Цикл үшін итератор циклы Linux циклы ретінде болып табылады. Бұл тек мәндер тізімін іріктейді. Формат келесідей көрінеді:

опция үшін % Var пәрменді жасау (тізім)

Келесі мысалда тізімдегі мәндерді қосатын болады. Ол цикл ішінде параметрлерді қолданбайды.

```
for %i in (1 2 3) do set /a sum=sum+%i
```

Тізімді бергеннен басқа директориялар секілді осындай пәрмендерді көрсетуге болады, сондықтан цикл үшін dir пәрменінен нәтижені іріктейді. Ол үшін цикл үшін-ге / п опциясын қосу қажет. Мысалы, келесі шығару бір орындағанда біреуден DIR пәрменін орындаған кезде ізделіп отырған позицияларда болады.

```
for /f %file in ('dir') do echo %file
```

Өкінішке орай, цикл үшін пәрменінің (' dir ') көмегімен комбинациялай отырып, жұмыс істемеуі мүмкін, себебі, dir пәрмені 'элементтердің бір-неше бағандарының құндылығын қамтамасыз етеді және эхо тек өрнектер әрбір сызықтың бірінші элементіне (бағанға) көтеріледі (соңғы модификациясының күні болып табылады). dir / б пәрмені элементтердің аттарын ғана береді. Жоғарыдағы нұсқауларды өзгерту үшін, F% файл /

('реж / б')-дағы эхо % файлды жасау.

Linux секілді DOS пәрмендері DOS қабығының сценарийлеріне орналастырылуы мүмкін. Бұл del (жою), move, copy, ren (атын өзгерту), dir, type (Linux' cat ұқсас), print (принтерге жіберу), Rmdir, MKDIR, CD және CLS (мөлдір экран) секілді пәрмендерді қамтиды. Сондықтан сіз сценарийлерді жаза аласыз, мысалы, файлдық жүйеде файлдарды тексеру үшін және егер файлдар болса немесе белгілі бір атрибуттарға ие болса, оларды жою, атын өзгерту немесе орнын ауыстыру.

Әрі қарай оқу

- Блюм, Р. және Бреснахан, С. Командалық жол Linux және Інжілдің Shell Scripting. Нью-Джерси: Wiley және Sons, 2011.
- Dietel, П. және Dietel, Н. С ++: Қалай бағдарламалау керек. Жоғарғы Saddle River, Нью-Джерси: Prentice Hall, 2011.
- Стекло, Р. Бастапқыда: Software Пионерлерінің жеке естеліктері. Нью-Джерси: Wiley және Sons, 1997.
- Kelley, А. және Pohl, И. С туралы кітап: С тілінде бағдарламалау. Reading, MA: Addison Wesley, 1998.
- Керниган Б. және Ритчи Д. С бағдарламалау тілі. Englewood Cliffs, NJ: Prentice Hall, 1988.
- Лян, Ю. Java-бағдарламалауға кіріспе. Boston, MA: Pearson, 2013.
- Митчелл, Ж. Бағдарламалау тілдеріндегі тұжырымдамалар. Ұлыбритания: Кембридж, Cambridge University Press, 2003.
- Ньюхем, С. Bash қабығын зерттеу: Бағдарламалау бойынша Unix Shell. Массачусетс: O'Reilly, 2005.
- Редди, Р. және Циглера, С. FORTRAN 77 с 90: Ғалымдар мен инженерлерге арналған қосымшалар, Сент-Пол, Миннесота штаты: West Publishing, 1994.
- Robbins, А. және Биб, Н. Classic Shell Scripting. Массачусетс: O'Reilly, 2005.
- Себеста, Р. Бағдарламалау тілдерінің тұжырымдамалары. Boston, MA: Pearson, 2012.
- Такер, А. және Нунан, Р. Бағдарламалау тілдері: қағидалары мен парадигмалары. Нью-Йорк: McGraw Hill, 2002.
- VENIT, С. және Субраманьян, П. Turbo Pascal-мен құрылымдық бағдарламалауға прожектор. Иган, MN: West Publishing, 1992.
- Уэббер, А. Заманауи бағдарламалау тілдері: Практикалық кіріспе. Wilsonville, OR: Франклин, Бидл және Associates, 2003.
- Weitzer, М. және Ерш, Л. MS-DOS / PC DOS түсіну және қолдану. Сент-Пол, Миннесота: West Publishing, 1986.

15. Ақпарат

Оқулық бірінші кезекте технологияны көрсетті. Бұл тарауда ақпараттық жағына бас назар аударылады. Соның ішінде, тарауда ақпарат түрлері, сақтауды ұйымдастыру және қайта өңдеу туралы ақпаратты қолдайтын бағдарламалық қамтамасыз етумен бірге қолдану қарастырылады. Со-сын осы тарауда ақпараттық қауіпсіздік пен сенімділік қарастырыла-ды: олардың ақпараты қорғалуды қамтамасыз ету үшін ұйымдастыру қабілеті. Осы мақсатта тарау ақпараттық қауіпсіздіктің технологиялық әдістерімен қатар ұйым қорғауы тиіс шабуалдардың көптеген форма-ларын сипаттайды. Тарауда шифрлауды қоса алғанда бағдарламалық шешімдер мен RAID-ті сақтау секілді аппараттық шешімдер қарасты-рылады. Тарау азаматтардың жеке өмірінің қол сұғылмаушылығы мен ақпараттық технологияларды қорғауға көмектесу үшін құрылған заңна-маға көзқараспен аяқталады.

Осы тарауды оқытудың мақсаттары:

- Ақпарат Ақпарат-Білім-Тәжірибе иерархиясына енгізумен келетін әр түрлі формаларды талқылаңыз.
- Деректер қорларын, деректер қорларын басқару жүйелерін және де-ректер қоймаларын сипаттаңыз.
- Ақпараттық қауіпсіздік пен қамтамасыз ету облысына тәуекелді тал-дау кезінде қолданылатын процессорларды енгізу.
- Ақпаратты қорғау облысындағы ақпарат пен шешімдер қауіпін талқы-лаңыз.
- Шифрлау технологиясына сипаттама беріңіз.
- АҚШ аумағына ақпаратты қолдануды жеңілдететін және тұлғалардың құпиялылығын қорғайтын заңнаманы енгізу.

Осы уақытқа дейін оқулықта біз негізінен ақпараттық технологияның (АТ) технологиялық жағына тоқталдық. Біз компьютерлік жабдықты, бағдарламалық қамтамасыз ету, операциялық жүйелер, бағдарламалау және алда олардың қолданылуы мен функциялары қарастырылатын

негізгі теориялардың бірқатарын қарастырдық. Бұл тарауда біз АТ-ның басқа жағы, ақпараттық жағын қарастырамыз.

Ақпарат деген не?

Ақпарат көбінесе өңделген деректер ретінде анықталынады. Яғни, өңделмеген деректер кірме болып табылады және ақпарат қандай да бір үдерістің шығуы болып табылады. Негізінде, жалпы біз интерпретацияланатын деректердің кез келген формасы ретінде ақпарат туралы ойлауымыз қажет. Бұл деректер ұйымдастырылуы, манипуляциялануы, сүзгіленуі, сұрыпталуы және есептеп шығару үшін қолданылуы мүмкін. Сосын бұл ақпарат адамға шешімдерді қабылдауды қолдау үшін берілген. Біз деректер мен ақпаратты жинаққа, сондай-ақ, формулалар мен рецептер, кітаптар мен әдебиеттердің басқа формалары, жоспарлар мен құрастырылмдар, сондай-ақ, компьютерлік бағдарламалар мен стратегиялар секілді артефактардың адамдық өндірістерін қамтитын ұйымдастырылмаған шығарулар интеллектуалды меншікке сәйкес келетін бастапқы деректерден өзгертін спектр түрінде қарай алар едік.

Алайда, АТ көзқарасы тұрғысынан ақпарат көбінесе АТ-үдерістердің соңғы нәтижесі болып табылатын, біз келесі сұрақтарды толығырақ қарастыруымыз тиіс:

- Ақпарат қалай сақталуы тиіс
- Бұл ақпарат қалай берілуі тиіс (немесе егер)
- Оны ақпаратқа айналдыру үшін деректерге қолданғымыз келетін үдерістер қандай?
- Бұл ақпаратты қалай қарау /визуалдандыру тиіс
- Деректер мен ақпараттың дәлдігін қамтамасыз етуге қойылатын талаптар қандай?
- Деректер мен ақпараттың қолжетімділігі мен қауіпсіздігін қамтамасыз етуге қойылатын талаптар, ақпараттың өзі ақпаратты өңдектің қалаулы соңғы нәтиже болып табылмайды. Информатика саласындағы зерттеушілер "DIKW иерархиясында" жасады. Бұл иерархия деректердің ақпаратқа, білімге, ақылға ауысуын анықтайды. Иерархияның әрбір деңгейі алдыңғы деңгейдің терминдерінде анықталады.

Деректерді адам (немесе компьютер) алды. Деректер әдеттегідей сигналдар секілді ойлайды. Адам үшін бұл сезімдер мүшелерінің (көру, дыбыс, иіс сезу, дәм сезу, сезім түйсігі) көмегімен алынған сигналдар болуы мүмкін. Компьютер үшін деректер қосымшаны іске қосқан кезде пайдаланушы енгізген мәндер болуы мүмкін, бірақ заманауи есептеу техникасында деректер камераның, микрофонның, штрих-кодты салы-

стырып оқығыштың, тетіктің немесе планшеттің көмегімен алынуы мүмкін. Кірме деректер әдеттегідей тиісті формаға түрленгенге дейін қолданыла алмайды. Екілік көзқарас болатын компьютер үшін.

Иерархияда ақпарат келесі деңгейде. Ақпарат көбінесе иерархияда деректерден алынғандай жиі анықталынады. Шығару термині (немесе шығару) бір немесе бірнеше үдерістер секілді оны неғұрлым қолайлы формаға түрлендіретін деректерге қолданды дегенді білдіреді. Кез келген үдерісте алынған ақпаратты неғұрлым құрылымданған әрі өзекті болып табылады деп қарастырылады. Идея деректердің тікелей қолдану үшін адамдарға арналған өте төмен деңгей болып табылатындығында көрінеді. Нөмірлер тізімі өздігінен шешімдерді қабылдауда пайдалы бола алмайды, ал нөмірлер тізімі статистикалық талдаудың көмегімен өңделеді, ол бізге неғұрлым маңызды нәрсені айтатын орташа мәнді, медиананы, стандартты ауытқуды және дисперсияны қамтамасыз ете алады.

Білім кез келген ақпарат немесе деректерге қарағанда неғұрлым тиянақсыз түсінік болып табылады. Білім кейде қолдануға енгізілген ақпарат секілді анықталынады. Басқа жағдайларда білім ақпараттың бірнеше әр түрлі көздердің синтезін білдіреді. Білімді контекстке орналастырылған ақпарат деп қарастыратын тәсілдердің бірі ақпаратты қолдану арқылы алынған тәжірибенің нәтижесі болып табылады. Сонымен қатар, біз білім туралы жетілдірілген ақпарат туралы ойлай аламыз, осылайша, пайдаланушы қажет болған кезде ақпараттың тиісті бөлігін ғана шақыруға қабілетті. Мысал ретінде алгебралық заңдарды оқып зерттеу жолымен алгебралық тапсырмаларды шешуге үйренуге болады. Заңдар ақпаратты білдіреді, ал осы жағдайда заңды тиісті қолдануды таңдаудың тәжірибесі білім болып табылады.

Ақырында, ақыл білімдерге арналған әлеуметтік мақсатты қамтамасыз етеді. Кейбіреулері ақылға "неге" түсіну ретінде қарайды – бұл "неліктен олар жасағандай болды" немесе "неліктен адамдар өздерінің жасайтындарын жасайды". Ақыл білім мен тәжірибе болған жағдайда ғана келуі мүмкін. Басқа жағынан, білімді басқа адамдардың тәжірибесінен алуға болады. Ақыл заманауи әлемде функцияға адамның қабілетін жақсартатын және нанымдар мен құндылықтарды назарға алатын негізделген шешімдерді қабылдайтын адам үшін соңғы мақсат болып табылады.

Кейбір жағдайларда түсіну термин иерархияға қосылады. Әдеттегідей, егер түсіну орнатылған болса, білім мен ақыл арасында орналастырылатын болады. Басқалары түсіну ақылдың бөлігі болып табылады деп қарастырады (немесе білім мен ақыл секілді деңгейлер үшін талап етіледі).

Келесі бөлімде біз деректерге және сақтаудың неғұрлым жиі таралған формалардың бірі – деректер қорларына тоқталамыз. Біз сондай-ақ Data фурнитура үйлерінде қолданылатын қосымшалар мен деректер қоймаларын да қарастырамыз. Сосын біз АТ саласындағы екі негізгі мәселелі салаларды қарастырамыз: ақпаратты қамтамасыз ету және ақпараттық қауіпсіздік. Әр түрлі мәселелердің шешімінің ішінде тәуекелдерді басқару, апаттарды жоспарлау, компьютерлік қауіпсіздік және шифрлау болады. Біз осы тарауды ақпараттық технологияларды қолданумен байланысты тиісті заңнаманы қарастырудан бастаймыз.

Деректер қорлары

Ақпарат әр түрлі формалардың жиынтығына ие. Бұл мәтіндерге жазылған ақпарат болуы мүмкін. Бұл адамдар немесе топтар арасындағы ауызша қатынас болуы мүмкін. Ол басқа адамдарға берілуі тиіс ойлар да болуы мүмкін. Дегенмен, мұнда ақпарат компьютерде сақталынады және сондықтан компьютер сақталынуы және өңделуі мүмкін. Сақтаудың бинарлы формасынан бөлек біз ақпаратты қолайлы формада ұйымдастырғымыз келеді. Яғни, біз сақталынып тұрған ақпараттың шынайы эквивалентін (баламасын) үлгілеуді қалаймыз. Әдеттегідей ақпарат жазбалар түрінде ұйымдастырылған және деректер базаларында сақталынады.

Деректер базасы деректердің ұйымдастырылған жиынтығын білдіреді. Біз деректер қорындағы деректерді деректер қорларын басқару жүйесінің (ДҚБЖ) қандай да бір формасының көмегімен жасаймыз, манипуляциялаймыз және қолжетімділік аламыз. ДҚБЖ Microsoft Access, SQL және оның вариацияларын (MySQL, Microsoft SQL, PostgreSQL және т.б.) және Oracle-ді қамтиды.

Деректер қорын ұйымдастырудың қарапайым қағидасы қатынас болып табылады. Қатынас әрбір жазба атрибуттар мәндеріне бөлінетін жекелеген жазбалар арасындағы қатынастарды сипаттайды. Мысалы, қатынас студенттердің жазбаларынан тұруы мүмкін. Әрбір студент үшін бірінші аты, тегі, студенттің ID, ірі, кішігірім (қалау бойынша) және орташа балл бар.

Қатынастың өзі кесте түрінде жиі көрсетіледі (15.1 кесте). Кестенің жолдары жазбаны білдіреді: қызығушылық танытатын жеке тұлғалар адамдар болып табылады ма, заттар немесе түсініктер (мысалы, жазба клиентті, автомобильді немесе колледж курсын сипаттауы мүмкін). Формалды түрде жазбалар (жолдар) кортеждер деп аталады. Қатынастың барлық жазбалары бірдей атрибуттар. Ол өрістер деп аталады және олар қатынас бағандарын құрады. Әдеттегідей, өріс сан, жол, күн,

иә / жоқ (шын өтірік немесе /) секілді деректер түрлерін сақтайтын болады, сондай-ақ, түрдің шегінде ерліктер болуы мүмкін, мысалы, сан ұзын бүтін, қысқа бүтін, затты сан немесе доллардағы сома.

15.1 кестеде біз студенттік академиялық қатынасты көреміз. Осыған байланысты, студенттік жазбалар алты өрісті қамтиды: жеке куәлігі, тегі, аты, мажор, минор және GPD. Жазбалардың әрқайсысы GPD үшін жол (әріптерді, тыныс белгілері мен сандарды біріктіретін мәтін) болып табылады. Студенттің ID-і бірнешеу болуы мүмкін, бірақ біз әдеттегідей бірқатар арифметикалық операцияларда қолданылуы мүмкін деректерге арналған сандарды қолданамыз. Себебі, біз студенттердің ING көп зарядты сәйкестендіргіштерді қоспайтындықтан, біз оларды жолдар түрінде сақтайтын боламыз.

Қатынас өрістері, бәлкім, сақталынуы мүмкін мәндерге қойылған шектеулер. Мысалы, GPA 0.000-4.000 ауқымы шегінде болуы тиіс. Біз заң мамандықтарының тізімі мен универсиетте кәмететке толмағандарға сәйкес келу үшін мәндерді мажор мен минорда шектеуімізге болатын еді. 15.1 кестесіне қатысты бір жазба үшін Минорға арналған ешқандай мән жоқ екендігіне назар аударыңыз. Біз барлық жазбалардың Минорды емес, майорға ие болғандығын талап ете алатын едік.

Қатынасты ескере отырып, біз "CIT майорлары бар барлық студенттерді көрсету", "GPA ≥ 3.0 болатын CIT майорлары бар барлық студенттерді көрсету", "GPD ≥ 3.0 болатын CIT майорлары немесе негізгі CSC болып табылатын барлық студенттерді көрсету" секілді бірқатар ақпаратқа арналған бірқатар ақпарат үшін деректер қорын сұратуды қалар едік. Сұратулардың бұл түрлері шектеулер секілді танымал. Осындай сұратудың нәтижесі берілген критерийлерге сәйкес келетін қатынасқа сәйкес жазбалардың тізімі болып табылады. Яғни, біз берілген критерийлердің негізінде кіші топқа қатынаспен шектелеміз. Біз сүзгілеу немесе іздеу секілді операцияның осы түріне жүгінуіміз мүмкін.

Сұратулар деректер қорынан деректерді қайтарады, дегенмен оларға шектеулер қойылмауы тиіс. Яғни, қайтарылатын жазбаларбен шектелмеуі тиіс. Проекция арақатынастан барлық жазбаларды қайтаратын сұратудың басқа формасы, бірақ атрибуттарды немесе өрісті ғана таңдау. Мысалы, біз тек бірінші ат пен тегін қамтамасыз ету үшін 15.1 кестеден арақатынасты жобалай аламыз.

Қатынас бойынша тағы бір операция бірқатар өріс(тер)ге негізделген жазбаларды сұрыптау болып табылады. Мысалы, 15.1 кестеде тегі бойынша, сосын аты бойынша өсу ретімен сұрыптау (осылайша, Ян Андервуд Рут Андервудтың алдында тұрады).

15.1 кесте. Байланыстардың деректер қорының мысалы

Студенттің ID	Аты	Тегі	Кафедра	Қораштық	GPA
11151631	George	Duke	MUS	PHY	3.676
10857134	Mike	Keneally	MUS	CHE	2.131
19756311	Ian	Underwood	MUS	HIS	3.801
18566131	Ruth	Underwood	MUS	MAT	3.516
18371513	Frank	Zappa	MUS		2.571

Біз қою операцияларын (жазбаны қосу), операцияларды жаңартуды (бір немесе бірнеше жазбалардың бір немесе бірнеше атрибуттарын өзгерту), операцияларды жоюды (бір немесе бірнеше жазбаларды жою) орындай аламыз. Жаңартумен бірге сүзгілеудің бірқатар критерийлерін немесе жоюды қолдана отырып, біз белгілі бір критерийлерге сәйкес келетін барлық жазбаларды өзгерту немесе жоя аламыз. Мысалы, қолы ашық мұғалім бола отырып, мен барлық CIT ірі компаниялар жоғары орташа балға ие болуы тиіс деп шешуім мүмкін еді, сондықтан жаңарту операциясы $GPA = GPD * 1.1$ көрсетуі тиіс, мұнда Major = "CIT". Яғни, әрбір жазба үшін негізгісі "CIT" болып табылады, GPA өрісі $GPD * 1.1$ жаңарту.

Деректер қоры көпшілігі жабатын бөліктерге ие қатынастар тобынан тұратын болады. Осылайша, ақпарат бір уақытта бірнеше қатынастардан алынуы мүмкін. Мысалы, біз студенттің байланыс ақпаратынан тұратын қатынасқа ие болуымыз мүмкін. Осылайша, 15.1 кестесінде көрсетілген арақатынасты студенттің схоластикалық ақпараты деп атауға болатын еді, ал студенттің байланыс ақпараты 15.2 кестеде келтірілгендей көрінуі мүмкін.

Екі арақатынасты ескере отырып, енді біз қосылыс деп аталатын деректер қорына басқа сұратуды қолдануымыз мүмкін. Біріктіру бірнеше қатынастардан ақпаратты кері сұратады. Мысалы, біз студенттің жеке куәлігін, тегін, майорларды, кәмелетке толмағандарды, GPD мекенжайды және телефондар нөмірін бере отырып, барлық студенттік жазбаларды алу үшін 15.1 мен 15.2 кестелерден екі арақатынасты біріктіре аламыз.

Біз және / немесе шығыңқы жерлердің және / немесе шектеулері бар жалғауды біріктіре аламыз. Мысалы, біріктіру мен проекция барлық студенттердің телефондар нөмірін және аты мен тегін бере алады. Бірік-

тіру мен шектеу Огайода тұратын СІТ майорлардың студенттердің барлық жазбаларын бере алады. Барлық үш операцияларды біріктіру, мысалы, Огайода тұратын барлық СІТ мамандықтардың бірінші мен тегін шығару.

Деректер қорында қолжетімді қосымша операциялар бар. Олар өзіне біріктіру, қиылысу және айырмашылық операцияларының жиынын қамтиды. Одақ көрсетілген қатынастардың ішіндегі барлық жазбаны біріктіреді. Қиылысу екі (немесе барлық) көрсетілген қатынастарда бар жазбаларды ғана шығарады. Айырмашылық екі (немесе барлық) көрсетілген қатынастарда болып табылмайтын жазбаларды қайтарады.

Деректер әр түрлі арақатынастарда қайталанатын атрибуттардың мәніне ие болатындығына назар аударыңыз. Мысалы, Огайодан үш студент бар, Андервуд тегі бар екі студент және Ақпараттық технологиялар мамандығында оқитын бес студент бар. Жұмыс істеу үшін кем дегенде деректер қоры үшін атрибуттардың бірі (өрісі) айрықша мәндерді қамтуы тиіс. Бұл өріс айрықша сәйкестендіргіш немесе бастапқы кілт секілді танымал. 15.1 және 15.2 кестелердегі екі қатынасы жағдайында айрықша сәйкестендіргіш студенттік билетті білдіреді. ДҚБЖ қосу операциясын қолдану кезінде әр түрлі арақатынастарда жазбаларды салыстыру үшін айрықша сәйкестендіргішті қолданатын болады. Мысалы, егер біздің сұрату $GPD \geq 3.0$ болатын студенттердің телефон нөмірін сұратса, ДҚБЖ GPD критерийлерге сәйкес келетін жазбаларға арналған схоластикалық ақпаратты іздеуі тиіс, сосын байланыс ақпаратынан телефон нөмірлерін шығаруы керек. Қатынастар арасындағы студенттерді анықтау үшін айрықша сәйкестендіргішті қолданады. Сонымен қатар

15.1 кесте. Басқа қатынастар деректер қоры

Student ID Адрес Город Штат Индекс Телефон

10857134 8511 N. Pine St Erlanger KY 41011 (859) 555-1234

11151631 315 Sycamore Dr Cincinnati OH 45215 (513) 555-2341

18371513 32 East 21st Apt C Columbus OH 43212 (614) 555-5511

18566131 191 Canyon Lane Los Angeles CA 91315 (413) 555-1111

19756311 32 East 21st Apt C Columbus OH 43212 (614) 555-5511

15.2 кестедегі жазбалар студенттік билет бойынша ретке келтірілетіндігіне назар аударыңыз (15.1 кестедегі қатынаста ретке келтірілген жазбалар секілді тегінен айырмашылық бойынша).

Реляциялық деректер қоры деректер қорларын ұйымдастыруға арналған бір ғана форматты, бірақ сөзсіз неғұрлым таралған форма болып табылады. 1980 жылдан бері зерттеушілер деректер қорларының басқа

формаларын зерттеді. Неғұрлым қызықты форматтардың кейбіреулері осында келтірілген.

- Белсенді деректер қоры сұратулар мен деректер қорларымен басқа операцияларға ғана жауап беретін статикалық реляциялық деректер қорына қарағанда деректер қорына қатысты іштей немесе сырттай туындайтын оқиғаларға жауап береді.

- 12 тарауда сипатталынғандай, деректер қорының бұлтты желілік қойма мен объектіні өңдеуді білдіреді, сондықтан, бұлттың деректер қоры бұлтта болатын (сақталынатын) қарапайым деректер қоры, осылайша, қашықтықтағы режимде қолжетімді.

- Таралған қор – деректер қоры бір орында ғана сақталмайды. Деректер қорының бұл формасы бұлт пен желілік деректер қорларын жабатын болады, деректердің параллельді қорын да жабуы ықтимал.

- Құжаттардың деректер қорлары, азырақ құрылымданған сұратуларға негізделген ақпаратты іздеуді орындайтын жазбалар (мысалы, квебек Райстың кілт сөзі, іздеу жүйесіне енгізілген), құжаттардың деректер қорлары кітапханаларда, іздеу жүйелерінде жиі кездеседі, мысалы, Google құжаттардың деректер қоры ретінде қарастырылуы мүмкін.

- Кіріктірілген деректер қоры – деректер қоры және ДҚБЖ деректер қоры қолданатын үлкен қосымшаның бөлігі болып табылады. Мысалдардың бірі бірнеше компоненттері бар медициналық диагностикалық эксперттік жүйе болып табылады: интерфейстің табиғи тілі, білім қоры, наукастар жазбаларының деректер қоры.

- Иерархиялық қор – деректер ата-ана-ұрпаққа үлгіленетін деректер қоры. Жазбалар деректер қорының бір жерінде орналасқандай, ал басқа бөлігі басқа салада орналасқандай бөлінген. Мысалы, деректер қоры қызметкер жасай алады, ең жоғарғы деңгейде, барлық қызметкерлердің тізімі. Бала қатынасы сосын осы жұмыскердің жағдайы туралы ақпаратты (мысалы, еңбекақы деңгейін, міндеттемелер) қамтуы мүмкін және басқа еншілес қатынас жұмыскердің жобалары туралы ақпаратты қамтиды.

- Гипермедиа деректер қоры – БАҚ гипер сілтемелері арасында байланысқан жазбаларды қамтитын деректер қоры. World Wide Web гипермедиа деректер қоры ретінде қарастыруға болады, мұнда "records " құжаттар. Бұл дәстүрлі емес деректер қоры, жазбалар қандай да нақты модада ұйымдастырылмаған, ал қатынастар осындай жазбаларды жинау үшін ұйымдастырылған болып табылмайды.

- көп шамалы деректер қорлары деректер қорының қатынасы әдетте кестелер түрінде саналады: екі шамалы көзқарастар. Көп шамалы деректер қорлары екі өлшемнен көп өлшемнен тұратын қатынастарды сақтайды.

Осындай қатынас үшінші өлшем уақыттың өзгеруін білдіретіндей деректерді ұйымдастырады. Мысалы, бізде топ болар еді.

15.1 кестесіндей қатынастар, мұнда әрбір қатынас нақты семестрде оқитын студенттерді білдіреді. Біз осы қатынастарды бірге біріктіргіміз және семестр өрісін қосқымыз келгенмен, үш шамалы арақатынас деректерді көру үшін неғұрлым қисынды тәсілмен береді, себебі, ол неғұрлым қисынды ұйымдастыруды қамтиды.

- Желілік деректер қоры таралған деректер қорлары секілді емес; оның орнына желі термині бір-бірімен бағдарланған граф секілді байланысқан деректер жиынын береді (графа математикалық түсінігін қолдана отырып). Желінің деректер қоры иерархиялық деректер қорына бірқатар ұқсас.

- Объектіге бағдарланған деректер қоры объектілерді анықтамайды, өзгермелі қатынастарға орналастырылған жазбалар секілді, объектінің деректері біртұтасқа жиналады. Сондай-ақ, объектіге бағдарланған бағдарламалауға байланысты объект объектіде сақталынған деректерге қолжетімділік пен оларды басқару үшін белгілі бір операциялармен қапшықтанады. Мәні бойынша, бұл деректер қорлары мен объектіге бағдарланған бағдарламалаудың үйлесімі.

Параллельно база-это база данных работает на с несколькими процессорами параллельно. Для того, чтобы гарантировать, что данные не повреждены (например, изменении на один процессор в то время как другой процессор использует одни и те же данные), синхронизация должна быть реализована на данных.

- Пространственная или временная база данных, база данных, которая хранит пространственную информацию, временную информацию, или обоих. Рассмотрим, например, медицинские документы, которые содержат записи о пациентах как последовательности событий. Специализированные запросы могут быть использованы для просмотра последовательности. Возможно, события 1 происходят до события 2? Или же они перекрывают либо соответствовать одному и тому же периоду? Пространственная база данных включает в себя возможность запроса о двух- и трехмерных взаимодействий.

Деректер қоры белгілі бір түрде ұйымдастырылған жазбалар жиынын білдіреді. Реляциялық деректер қоры қатынастарды қолданады. Жоғарыда көрсетілгендей, басқа ұйымдастырулар өзіне объектіге бағдарланған, иерархиялы, гипермедиа, желі, кеңістікті және уақытшаны қамтиды. Деректер қоры бірнеше жазбалардан немесе миллионнан тұруы мүмкін. Деректер жиыны неғұрлым көп болса, сыни ұйымдастыру соғұрлым

айқын түсініп, көрді. Деректер қорлары олардың өлшемдері мен күрделілігінде өсетін болса, біз осы деректер қорын деректердің қарапайым жиыны ретінде емес, деректер қорларының жиыны ретінде қарастыруға бейімбіз. Ескі ұйымдастырылған деректер қорларының жиыны деректер қоймасы деп аталады. Қарапайым деректер қоймасы ETL үдерісін қолданады.

- Деректерді алып шығару - деректер әр түрлі көздерден келіп түседі, бұл көздер сәйкестендірілген, түсінікті болуы тиіс.
- Деректер әр түрлі көздерден болу шартымен деректерді түрлендіру, деректердің сол атрибуттардың жиынын қолдануымен ұйымдастырылмауы әбден мүмкін. Осылайша, деректер деректер қоймасының үлгісіне (үлгілеріне) сәйкес келуі үшін түрлендірілуі тиіс. Бұл өзіне тиісті атрибуттарды (өрістің) тандай отырып, айрықша сәйкестендіргішті мойындайтын қатынастардың деректерін немесе объектілерді (қайсының форматы қалаулы болатынына байланысты) қамтитын болады.
- oadL-дағы деректер қойма, яғни түрлендірілген деректер деректер қорында көрсетудің тиісті форматын қолдана отырып, сақталынуы тиіс.

Бизнес үшін деректер қоймасына арналған деректер көздердің кез келген санынан шығуы мүмкін.

Оларға келесілер жатады:

- Кәсіпорынның ресурстарын жоспарлау жүйелері
- Жеткізу тізбектерін басқару жүйелері
- Маркетинг және қоғаммен байланыс есептері
- Сатылым рекордтары
- Сатып алу және инвентарлық есепке алу
- Клиенттен тікелей енгізу (мысалы, вебформадан немесе тексеруден кіру)
- Адам қатынастарының жазбалары
- Бюджетті жоспарлау
- Жобалар бойынша ағымдағы есептер
- Бухгалтерлік есептің жалпы деректері

Деректер жиналып, түрлендірілгеннен кейін, олар деректер қоймасында сақталынады. Енді, деректерді қолдануға болады. Қолдануға болатын әдістер бар, ал нәтижелер өңделген және төменде суреттелгендей ұйымдастыру білімдерін әрі қарай жылжытуға арналған қоймаға кері сақталынуы мүмкін. Бұл әдістер кейде пайдаланушылар қоймадан деректерді немесе ақпаратты ала алатын құралдар деректері бар витрина-

лар деп те аталады.

Операциялардың деректер қоймасы деректер қорларын басқарудың қарапайым операцияларынан талдау мен статистиканың неғұрлым күрделі алгоритмдеріндегі ауқымда. Дәстүрлі ДҚБЖ көзқарасы тұрғысынан, пайдаланушы сіздің сатып алуыңызды, инвентарьді, сатылымдарды және жұмыс күшіндегі болжанатын қажеттіліктерді түрлендіруге арналған бюджеттік деректерді оқып зерттей алады. Немесе сатылымдар мен қоғаммен байланыс бойынша есептер қандай өнімдерді болашақ маркетингтік стратегиялар есебінен көрсету қажеттілігін анықтауға көмектесуі мүмкін.

OLAP-да (Online Analytical Processing) деректер құралдарды талдау үшін бағдарламалық қамтамасыз ету жиыны арқылы өңделеді. Сіз OLAP секілді ДҚБЖ жоғарғы бөлігінде отырып, елестетуіңізге болады, сондықтан, пайдаланушы деректер қорынан деректерді алып шығара алады және талдаудың неғұрлым күрделі операцияларынан ДҚБЖ операцияларын бөлу қажеттілігінсіз деректерді талдай алады. OLAP операциялары деректерді категорияландыру, қайта араластыру, деректер детализациясының деңгейін төмендету, жинап қою (бір жаққа бұрып қою) және бұруды қамтиды.

Бойлық деректерді алу бір өлшемдегі бірнеше өлшемдерден деректерді азайту жолымен деректердің кіші жиынтығын құрайды. Мысалы, егер біз біздің деректеріміздегі үш өлшемдегі деректер туралы ойласақ, Бойлық деректерді алу деректердің бір шамалы ұғымды құрады. Көлденен деректерді алу тек нәтиже бірнеше өлшемдерде болғаннан басқа Бойлық деректерді алу секілді, бірақ қазірге кезге дейін деректердің кіші жиынтығын алады. Мысалы, бірқатар жазбалар мен өрістерді тастау жолымен деректерді кіші үш шамалы жинауға деректердің үш шамалы жинауын шектеуі мүмкін.

Категориялау және деректер детализациясының деңгейін төмендету деректерді ұсынуды ығыстырады. Деректер детализациясының деңгейін төмендету көбірек детальдармен, бұрғылау жинақ деректерді береді. Жинап қою (бір жаққа бұрып қою) деректерді жинақтауына байланысты категориялауға ұқсайды, бірақ ол бірнеше элементтерден (мүмкін бір өлшемнен көп) деректерді бір мәнге бұзады. Мысал ретінде, барлық HR жазбалар ағымдағы қызметкерлердің саны немесе әрбір позициядағы (мысалы, басқару, техникалық қолдау) қызметкерлердің санын білдіретін бір вектордың саны ретінде бір нөлдік нүктеге жинап қойылуы мүмкін.

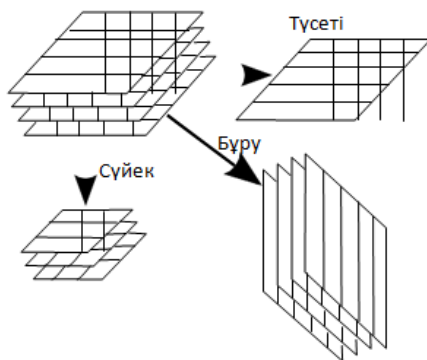
Ақырында, басқа көзқарас тұрғысынан деректерді қарауға арналған деректердің айналу осі. Жылдар және елдер бойынша өнімді сату туралы

акпаратты қамтитын деректер қорын қарастырамыз. Егер деректердің типтік түрі өнім болып табылса, біз оның орнына барлық деректерді әрбір жыл көрсететіндей етіп бұра аламыз. Негіз сосын жаңа көзқарас тұрғысынан деректерді қайта ұйымдастырады. Біз осы деректерді сату елдері бойынша бұра аламыз.

15.1 суреті категориялау, қайта араластыру және бұрылған болатын идеяларды көрсетеді. Мұнда бізде деректер жиыны бар. Бастапқы деректердің әрбір қабаты (немесе беті) басқа жылдың тұрғысынан қатынастың деректер қорын білдіреді. Мысалы, жоғарғы қабаты 2012 жылдан бастап клиенттер туралы жазбаларды қамтиды, келесі қабат 2011 жылдан бастап клиенттер туралы, ал төменгі екі қабат сәйкесінше 2010 мен 2009 жылдан бастап клиенттер туралы жазбаларды қамти алады. Кесік бір толық қабатты зерттеуі мүмкін, 2010 жылы бірнеше өлшемдердегі кіші жиынтық болуы мүмкін, мысалы, 2012, 2011 және 2010 жылдардағы деректерді, Оғайодан клиенттердің жазбаларын, тегі жолы, жұмсалған сома, сондай-ақ, жалпы кірген санын шектей отырып. Осы жылдар бойынша деректері емес, алдымен деректер мемлекет жағдайына бөлінетіндей, сосын әрбір жылға бөлінетіндей деректерді қайта ұйымдастыра аласыз.

OLAP талдауы 1960 жылдан бастап бар. Қазіргі кезде бәрінен де маңызды, себебі, деректердің қоймасы деректер қорларымен қарапайым операциялардан пайдалы жауаптар алу үшін өте үлкен болып табылады.

Деректер қоймасында қолданылуы мүмкін құралдардың тағы бір жиыны деректердің интеллектуалды талдауы болып табылады. OLAP-қа қарағанда, деректердің интеллектуалды талдауы салыстырмалы түрде жақындағы идея болып табылады, CIAL барлау зерттеулерінің трамағы ретінде 1990 жылдары пайда болды. Деректердің интеллектуалды талдауында идея әр түрлі статистикалық тұтас қатарды қолдануда көрінеді.



Талдаудың OLAP (Online Analytical Processing) әдістерін көрсету.

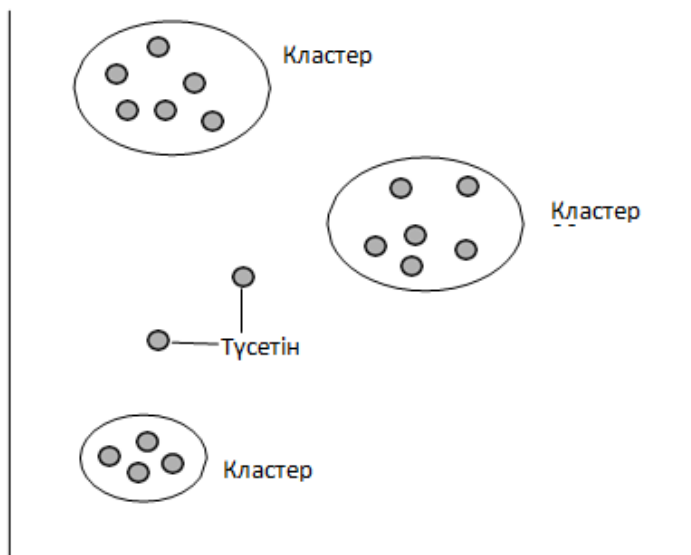
Деректерді жинау және егер нәтижелердің мәні болса, қарау операциялары. Адамның көмегімен қозғалысқа келетін OLAP-қа қарағанда деректердің интеллектуалды талдауының талпыныстары осы деректердің бірнеше кездейсоқ зерттеулеріне ие. Егер нәтижелердің мәні болса, алдын ала білу тәсілі жоқ. Деректерді интеллектуалды талдауда қолданылатын әр түрлі алгоритмдердің тұтас қатары бар. Мұнда біз кеңірек таралған әдістердің бірқатарын қарастырамыз.

Кластерлеуге деректер барлық өлшемдердің (өрістердің) емес, бірқатарының бойында топтасады. Өрісті ақылмен таңдағанда сіз деректердің бірнеше әр түрлі топтарға бірігетіндігін анықтайсыз. Егер сіз әрбір ұйысқанның мәнін анықтай алсаңыз, сіз өзіңіздің деректеріңіз туралы бірдеңе білдіңіз. 15.2 суретте сіз деректердің екі өлшемді қолдана отырып, топтастырылғандығын көре аласыз. Мысалы, егер осы деректер науқастар туралы жазбалардан тұратын болса, біз оларды жасы мен салмағы бойынша ұйымдастыра аламыз. Жақындық негізінде біз екі шеткі деректері бар үш кластерді анықтадық. Шеткі деректердің біреуі 2-топқа, басқасы 3-топқа жатуы мүмкін немесе екі дерек те олардың өздерінің топтарына жатуы ықтимал. 2-кластер жасы мен салмағына негізделе отырып, белгілі бір ауруды неғұрлым сезгіш науқастарды көрсетеді.

Кластерлерді автоматты түрде сәйкестендіру үшін компьютер жеке жазбалар арасындағы арақашықтықты анықтайды. Ол топтардың арасынан орталығын табады, сосын қай кластер бәрінен де жақын келетіндігін анықтау жолы бойынша деректердің басқа элементтерін қосады.

Кластерлерді қалыптастыру кезінде біз сіздің өрістердегі бастапқы таңдауыңыз бізге ешқандай пайдалы топтамаларды бермейтіндігін анықтауымыз мүмкін. Мысалы, егер біз жасы мен салмағының орнына бойы мен салмағын тандасақ, біз қандай да бір пайдалы сипаттамаларын анықтауымыз да мүмкін. Диабеттің 2-типіне сезімтал болуы мүмкін науқастарды анықтағымыз келеді деп ойлаңыз. Жасы, бойы және табыстар деңгейі секілді өрістер пайдасыз болуы мүмкін, Дегенмен, салмағы (немесе ең болмағанда дене массасының индексі), этникаға жатуы мен жұмыс орнындағы күйзеліс деңгейі неғұрлым маңызды кластерлерді табуы мүмкін.

Бұл мәселеге әкеледі. Жасын, бойын, салмағын, тіпті табысын ескере отырып, жұмыстағы күйзеліс пен этникаға жатуындағы секілді графикті оңай тұрғыза аламыз ба? Біз деректерді санаттардан сандық мәндерге түрлендіру тәсілін табуымыз қажет.



15.2 сурет. Деректерді кластерлеудің нәтижелері.

Кластерлеу деректерді талдаудың өте таралған әдісі болып табылады, бірақ бірқатар мәселелер бар. Біз қандай да бір пайдалы кластерлерді тапқанға дейін әр түрлі өрістердің үйлесімінің жиынтығы бойынша топтастыруға тырысыпы көруіміз керек шығар. Өрістердің санының артуына байланысты комбинациялардың саны геометриялық прогрессияда артады. Мысалы, 10 өрістерде тырысып көруіміз қажет 1024 әр түрлі комбинациялар болса, 20 өрістермен комбинациялардың саны 1 миллионға дейін артады.

Кластерлеудің бір түрі жақын көршілерді анықтауда көрінеді. Жақын көрші есептеп шығарылған кейбір математикалық теңдеуге бәрінен де жақын болатын деректердің элементі (мысалы, формуланың евклидік арақашықтығы). * Жақын көршінің алгоритмі арақашықтығы кейбір берілген санға қарағанда аз кластерлерді позициялау мен сәйкестендіруді жазу үшін кластерлеуде қолданылуы мүмкін. Алгоритм өрістердің бойындағы арақашықтықты есептеп шығарады. Тағы бір есептеп шығарылатын күрделі тапсырманы таңдау. Мысалы, егер 10 өріс болса және $k = 6$, біз жақын көршіні анықтау үшін 10-нан алты өрістің барлық мүмкін комбинацияларды қолдануға тырысар едік.

Ассоциацияның оқыту ережесі қызығушылық тудыратын қатынастарға

арналған деректерді іздеуді орындайды және оларды қамтамасыз етеді. Мысалы, азық-түлік дүкенінің 10000 түбіртектерін талдауда біз 4000-ы біреудің жержаңғақ майын, сондай-ақ, 4000-ның 3600 түбіртегі онымен қоса нанды атып алғандығын көрсетеді. Бұл ережеге айналады:

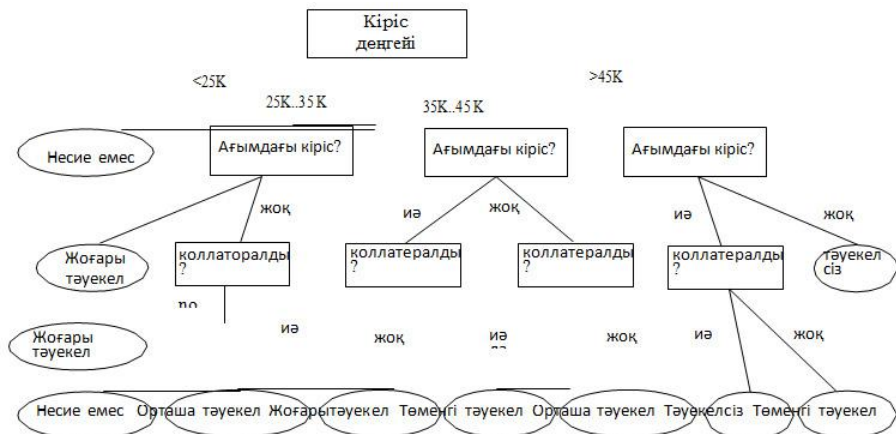
егер клиент жержаңғақ майын сатып алатын болса, онда ол нанды да сатып алады

Ереже 90% (4000-нан 3600) жиілігіне ие. Жиілік ереженің көбінесе шын болатындығын сипаттайды. Бірақ жиілік ереженің қаншалықты дұрыс болатындығын айтпады. 12 адам (10000-ның ішінен) суши сатып алғандығын және осы 12 адамның ақ шарапты да сатып алғандығын тапқанымызды ескеріңіз. Жиіліктің қаншалықты өте жоғары болғанына қарамастан (100%), ереже онша пайдалы емес.

Шешім қабылдауға көмектесу үшін ассоция ережесін қолдана аламыз, мысалы, маркетинг пен сауда саласында. Мысалы, біз жержаңғақ майын нан тұратын жерге қою туралы шешім қабылдай аламыз және біз "нанды сатып алу және 25%жержаңғақ майының банкасын алу" дей отырып, екі өнімді жылжыту үшін сатуды шеше алар едік. Ассоциативтік ережелерді үйрету нәтижесі маңызды болу-болмауы мүмкін. Суши ережесі бізді суши мен шарап туралы ерекше бірдеңе жасауға сендіре алмауы ықтимал. Немесе шарап сатып алатын адамдар сыра сатып алмайды деген басқа ережені қарастырайық. Осы корреляцияда мағына бар ма? Ассоциативті ережелерді үйретудің артықшылығы біз деректерді интеллектуалды талдау құралы қолжазбасна деректерді бере алатындығымыз болып табылады және оған біз үшін ережелерді табуды беріңіз. Біз жиілігі 80%-дан асатын ережені ғана қайтару үшін құрал деп айтар едік. Сонда, ережені қалай қолдану керектігін шешу үшін басшылыққа осыны айтар едік.

Деректерді интеллектуалды талдаудың басқа өнімі шешімдер ағашы болып табылады. Шешімдер ағашы жасы, жынысы және табысы секілді түйіндер ретінде деректер қорының өрісін қамтитын ағаш тәрізді құрылымды білдіреді. Ағаштың бұтақтары өрістердің әр түрлі ықтимал мәндерін білдіреді. Мысалы, егер түйін жыныс өрісіне ие болса, онда бұл

* Евклидтік арақашықтық – деректер арасындағы формула деректердің 1 және 2 а, b және c үш мәніне ие екендігін білдіреді.



Құтқару үшін деректерді интеллектуалды талдау

Интеллектуалды талдау қазіргі таңда басшылыққа шешім қабылдауға және болжауға көмектесетін бизнес құралы болып табылады. Алайда, мәліметтерді табу түскен пайдадан да ары пайдалы болу үшін құрылған. Міне мәліметтерді интеллектуалды талдауды пайдаланудың бірнеше сәті. Минессотаның енулерін байқау жүйесі компьютерлік жүйелердегі әртүрлі типтегі интрузия бола алатын ауытқуларды табу үшін, желілік трафикке жататын үлкен көлемді деректерді талдайды. Миннесота Университетінде 40000 астам компьютерлер бойынша тәжірибелер көптеген құрттар талпыныстарының бейімделуін тапты.

Биоинформатикада мәліметтерді интеллектуалды талдау адам геномына жататын мәліметтерді миллиардтаған бөлшектер арқылы електен өткізуге көмектесу үшін үнемі қолданылады. Соңғы мәлімдемелер ауру диагноздарының жақсаруына және емдеуді оңтайландыруға және гендердің өзара әрекеттесуінің ең үздік үлгісін жасауға әкелді. Сонымен қатар, BioSTORM мәліметтерді интеллектуалды талдау және індет пен биотеррорлық қақтығыстарды қадағалау үшін веб-ақпаратты өзіне біріктіреді. Ұлттық күзет агенттігі мәліметтерді интеллектуалды талдауды 1,9 трлн астам әйгілі лаңкестер иен азаматтардың арасындағы байланысын іздейтін телефон жазбаларына қолданды. Интеллектуалды талдаудың басқа әдістері Қара Web, лаңкестік ұйымдарға күдікті, құралдар тобы мен веб-сайттардағы сілтемелер құру үшін қолданылды.

Құқық қорғау ұйымдары мәліметтерді интеллектуалды талдауды өз қалаларындағы ыстық нүктелердегі мүмкін деген қылмыстарды ашу үшін, болашақ қылмыстың қай жерде болатынын болжау үшін, және бұрынғы қылмыстары бойынша қылмыскерлерді іздеу үшін пайдаланады. Іздеу

жүйелері мен ұсыныстық сайттар сіздің кіру сауалдарыңызды негізге ала отырып, қызығушылық тудыратын, сілтемелерді табу үшін үнемі мәліметтерді интеллектуалды талдауды пайдаланады. Мәліметтерді табуда Google мен Amazon пионер болып табылады. Бірі еркектерге арналған және бірі әйелдерге арналған екі тармағы болады (егер жалғау нүктесі секс үшін белгісіз немесе қол жетімді болмаса, үшінші тармағы болуы мүмкін). Жас шамасы сияқты алаң үшін, ықтимал мән үшін бұтақ ұсыну емес, маңызы бірге топтастырылған. Мысалы, (0 ден 12 жасқа дейінгі) жастағы жасөспірімдер жасы үшін бөлімшелер, жасөспірімдер жасында филиал, жас ересектер үшін бөлімшелер (мысалы, 20 дан 32 дейін), орта жастағылар үшін бөлімшілер (мысалы, 33 тен 58 дейін), сондай ақ зейнеткерлік жасындағылар үшін бөлімшелер (мысалы, 59 жас және одан жоғары) болуы мүмкін. Шешім ағаштарының жалпақ түйіндері біздің жасағымыз келген шешімдерді білдіреді. Сонан соң ағаштың сәйкес тармақтарына тиісті шешім қабылдау процесін білдіреді.

15.3 суреті шешім ағашының жалпы мысалын білдіреді. Бұрынғы несиенің банк шешімін есепке ала отырып, біз бұл процесті автоматтандырғымыз келеді. Бұл жағдайда біз шешім ағашын клиенттердің жазуларынан, және егер олардың кепілдіктері болса, ағымдағы қарыздары бар ма немесе жоқ па, олардың кіріс деңгейлерін есепке ала отырып құраймыз. Біздің ағашымыз үшін бұл үш алаң біздің түйіндерімізді (қораптарымен суретте көрсетілген) қамтиды. Сала әрбір алаң үшін ықтимал мәндермен мазмұндалады. Біз сынып қалдық кіріс деңгейі ықтимал төрт мәнге, "<25K", "25K ... 35K", "35K ... 45K", и "> 45K". Қалған екі алаң, ағымдағы және ілеспелі, жоқ / бар деген жауапқа ие. Жапырақ түйіндері бұрын қабылданған кредиттік шешімді қамтиды. Бұл түйіндер сопақшамен белгіленген. Мысалы, ағымдағы қарызы жоқ, кірісі 45 000 \$ артық адам ешқандай тәуекелсіз, ал кірісі 25 000 \$ кем адамға кез келген жағдайда несие беруге болмайды. Кірісі 25 000 \$ дан 35 000 \$ дейінгі, қарызы жоқ адам, несие алу үшін бұрынғыдай кепілді талап етеді, тіпті бұл жағдайда да жоғары тәуекелділік болып саналады. Шешім ағаштары дерек қорынан автоматты түрде өндіріле алады. Алайда алдыңғы мысалда тұспалданғандай, бірнеше мәннен де көп мәнге ие болса (мысалы жас, кіріс), онда мәліметтер алдымен саралануы тиіс. Бұл адамның немесе мәліметтерді интеллектуалды талдау алгоритмінің көмегімен жасалады. Кез келген жағдайда, шешім ағашын тудыру алгоритмі сондай ақ қарқынды шешімдермен, өйткені шешім ағашында қандай алаң қолданылатынын (15.3 суретіндегі шешім ағашында жынысы және жас мөлшері қолданылмады) және алаңдардың қолданылатыны үшін жасалады. 10 өрістер қатынасы дерекқорын ескерсек алаңдар

қолданылуы мүмкін әртүрлі 1024 комбинациялар, миллиондаған потенциалды ретке келтірілген алаңдарды айтпағанда(ағашта, үстінен астына), бар. Бүгін деректер қоймасы үлкен кәсіпорындар үшін деректерді ұйымдастыру мен қолданудың кең таралған түрі болып табылады. Деректер қоймасын корпорациялар, қаржы институттары, университеттер, емханалар мен медициналық мекемелер, телекоммуникациялық компаниялар, авиация саласы, барлау қоғамдастықтары және көптеген үкіметтік мекемелер пайдаланады. Бұлттық есептеулермен қатар(12 тарауда енгізілген), деректерді сақтау қоймасы деген көпшілік «Big Data» деп атағандықтан АТ масштабында ығысу дегенді білдіреді.

Деректер қоры мен қоймасын талқылағанда маңызды бір сипаттама жетіспейді. Деректер қоры мен қоймасының ақпараттық технологияны пайдалана отырып сақталатынына қарамастан, олар шындығында ақпараттық жүйе түрінде топтастырылады. Қандай айырмашылығы бар? Ақпараттық жүйе ақпаратты(деректерді) өңдеуді қамтамасыз ететін және сақтайтын жүйені қамтиды. Дегенмен, ақпараттық жүйе идеясы технологияға бейтарап болып келеді. Бүгін барлық ақпараттық жүйелер компьютерде сақталады және ақпаратты өңдеу үшін бағдарламалық жасақтама пайдаланылады деген шындық болғанмен, осылайша болмауы керек. Ақпараттық жүйе 3x5 карточкасында сақталатын немесе компания инвентаризациясының дерекқорында сақталатын кітапхананың картотекасы, қағазда картотекада сақталады. Шындығында, ақпараттық жүйе термині компьютер деген терминге қарағанда біраз ұзақ тіршілік етті. Дерекқор және мәліметтер қоймасынан басқа адамдар сараптама-лық жүйені және шешім қабылдауды қолдау жүйесін, географиялық ақпараттық жүйесін және корпоративті жүйені, ақпараттық жүйенің бір түрі ретінде жиі қарастырады. Ақпараттық техноглогия термині өздігінен ақпараттық жүйені қолдай алатын, сонымен қатар телекоммуникациялық жүйе, сенсорлы желі, ақпараттық жүйеден өзгеше, кейбір аттын өңделген үлестіруі сияқты көптеген басқа жүйелерді қолдай алатын технологиялық инфрақұрылымды сипаттайды. Біз 16 тараудағы мансаптың нұсқасы ретінде ақпараттық жүйені қысқаша қарастырамыз.

Қауіпсіздік және ақпарат

Ақпаратты қамтамасыз ету мен ақпараттық қауіпсіздік бірге қауіпсіздікті қамтамасыз ету ақпараты ретінде жиі сипатталады (IAS). Бұл бөлімде біз ХҚЕС жатқан тұжырымдаманы қарастырамыз. Біз ақпараттық қауіпсіздіктің кейбір бөлшектерін Қатер және шешім бөліміне қалдырамыз. IAS үлгісі үш компонент пайдаланатын IT-коммуникациясының, құрылғылардың және бағдарламалық қамтамасыз ету үш құраушысын біріктіреді. Назарды ақпараттың қалай пайда-

ланылатынына шоғырландыруға ХҚЕС үш мақсатқа тиым салады, әдеттегідей, ОББ атағандай: құпиялылық, тұтастық, қолжетімділік.

Құпиялылық деректердің қауіпсіздікте сақталғанын талап етеді, сондықтан бөтен, уәкілетсіз тұлғалар ала алмайды. Құпиялылық қауіпсіздік тетіктері шегінен тыс тұрады, себебі ол деректерді авторландырылған қолданушы болып саналатын саясат мәселелеріне қатысты, мәліметтер сақталған сияқты құпиялылық саясаты да сақталуы керек. Мысал ретінде осы тараудың басында біз ерекше идентификатордың қажеттілігі жөнінде талқылағанбыз. Ұйымда қызметкерлердің ерекше идентификаторлары ретінде әлеуметтік сақтандыру нөмірін пайдалана алу саясатын қолданады. Алайда, мұндай саясат олардың әлеуметтік сақтандыру нөмірін (жалақы есептеу үшін ғана қолжетімді болатын) қорғау үшін жұмысшының бостандығын бұзады.

Сонымен қатар қауіпсіздік механизмі компьютерлік желі арқылы рұқсатсыз қол жеткізуден қорғайтын механизмнен алда болуы керек. Мысалы портативті компьютерде мәліметтер қол жетімді емес немесе портативті компьютерлер бөтен тұлғалар үшін қол жетімді емес деген құпиялылық кепілдік беруге міндетті. Ноутбуктің ұрлануы ұйымның құпиялылығын бұзбауға міндетті.

Адалдақ мәліметтердің дұрыс болғандығын талап етеді. Ол үшін кем дегенде әртүрлі үш күш-жігер қажет. Біріншіден, мәлімет жинау жинаған мәліметтің дәлдігін қамтамасыз етуін қамтиды. Екіншіден, мәліметтер жүйеге дәл енгізілуі тиіс. Ең бастысы мәліметтерді түрлендіру қадағаланып отыруы тиіс. Яғни, мәліметтер дерекқорға енгізілгеннен кейін, кім, қашан, және қандай өзгерістер енгізілгені туралы, және ненің өзгертілгені жөнінде жазбалар қалдыру керек және қадағалануы тиіс. Бұл деректерге аудит жүргізіп отыруға және керек жағдайда мәліметтерді кері қайтаруға мүмкіндік береді. Сонымен қатар, байланыстыру нүктесін бір адам пайдаланса, екіншісінің өзгергенінен ИГД қауіпсіздікте болуға міндетті. Мәліметтердің тиісті сәйкестендіруінің болмауы, алдында осы тарауда талқыланды, мәліметтердің зақымдалуына әкеп соғады. Ақырында, ақпараттың қажет уақытында қол жетімді болуын талап етеді. Бұл айқын талап сияқты көрінгенмен, кейбір жағдайда шиеленістер құпиялылықпен байланысты сияқты, кіріктірілген қауіппен де байланысты. Ұйымдардың көпшілігі желілік сәнде оларды қол жетімді етеді деп ойлаймыз, сондықтан қолданушылар мәліметтерді ұйымда орналасқан компьютерден ала алады, сондай ақ ұйым алаңынан тыс қашықтықтан қол жеткізе алады. Құпиялылықты сақтау үшін, қауіпсіздіктің тиісті шараларын қолдану үшін, соның ішінде шифрлау/шифрды ашу, автори-

зациялаудың қорғалған нысаны және тиісті қолжетімділікті бақылау(-мысалы, файлдарға қол жеткізу құқығы) шаралары қолданылуы тиіс. сонымен қатар, егер бір пайдаланушы кейбір мәліметтерге қол жеткізсе, тұтастықты қолдау үшін, басқа пайдаланушылар бұл мәліметтерге қол жеткізе алмайды, осылайша олардың қол жетімділігіне шек қойылады. Тұтастыққа жәрдем ету мақсатында мәліметтердің заманауи резервті көшірілуі тиіс, бірақ әдетте, көшірілген файлдар сияқты олар қол жетімсіз. Қолжетімділік үшін қосымша қиындықтар болып кез келген компьютер, компьютерді апгрейдтеу кезінде пайда болған тұрып қалу уақыты, электр қуатының іркілістері, құрылғының іркілісі немесе қызмет көрсетуден бас тарту болып табылады.

2000 жыл

Мәліметтер тұтастығы лайықты мақсат болып табылады, бірақ бағдарламалық қамтамасыз ету тұтастығы қалай болады? Онда көптеген бағдарламалық қателіктер болды, алайда, Y2K артық белгілісі жоқ шығар. Y2K деген не, және неге бұлай болды? 1960 және 1970 жылдары бағдарламашылар үшін ол ортақ болған, жадыда барынша көп орын сақтау үшін, себебі компьютерлерде негізгі жады көлемі өте шектеулі болды. кеңістікті үнемдеудің бір амалы, бір жылға сақтауға арналған, кеңістікті шектеу болды. төрт мәнді санмен сақтамай, олар әдетте, екі мәнді сан ретінде сақтайды, мысалы 1969 жылдың 31 қаңтарда 01, 31 және 69 болып сақталды, мұның, шамамен 1997 жылға дейін мәселе болмайтынын ешкім түсінбеді.

2000 жылдан бастап қандай да бір мақсатты жылда ағымдағы жылдың салыстыруымен айналысқан бағдарламашылар кез келген логиканың кенеттен бұрыс болып кететінін түсіне бастады. Дауыс беру үшін адамның жеткілікті кәрі болғандығын анықтайтын келесі код бөлшегін қарастырайық:

Егер (CURRENTYEAR – туған жыл > = 18) ...

Енді, егер сіз 1993 жылы туған болсаңыз және осы шақта уақыт 2013 жыл болса не болатынын көрейік. Төрт мәнді жылды пайдалансақ бізде бары:

Если (2013 - 1993 > = 18) ...

Бұл шындық, сондықтан код дұрыс жұмыс істейді. бірақ енді, екі санмен не болатынын қарайық:

Егер (13 - 93 > = 18) ...

Бұл дұрыс емес, дауыс беру үшін сз жеткілікті ересек болсаңыздар да, дауыс бере алмайсыздар! ҮЗК адамдарға әлеуметтік қамсыздандыру чегін алуға кедергі келтіретіндіктен, есептеу үшін банктік бағдарламалық қамтамасыз етуде бұрыс қызығушылық тудырғандықтан, тіпті студентердің оқуын аяқтамауын болдыртқандықтан біздің қоғам үшін елеулі қатер туғызған.бірақ ешкім мәселенің шынайы аумағын білмеді, біз де білмес едік, егер ракеталық шахталар 2000 жылдан бастап өз зымырандарын кенеттен ұшыра бастамағанда. Электр компаниялары өздерінің клиенттеріне электр қуатын автоматты түрде өшіріп тастай ма? Медициналық құрылғылардың бұзылуы орын ала ма? Аспаннан ұшақтар құлай бастай ма немесе әуе басқару құрылғылары қозғалысы жабылып қала ма?

Сондықтан бағдарламашыларға екі мәнділерді төрт жылға дейін өзгерту үшін бар кодты өзгертуге тура келді.Бір Құрама Штатта ғана бұл әрекет 100 миллиард \$ артық тұрады.

Кейбір тиым салынған, бәрі емес, қосымша екі сала ХҚЕС түпнұсқа және кідіріссіз болып табылады.Бұл электронды саудада өте маңызды. Түпнұсқа деректердің, ақпараттың немесе қордың түпнұсқа екендігіне кепілдік береді.Бұл шынайы мәміленің орын алуына рұқсат етеді(жеке немесе ғаламтор арқылы). Түпнұсқалылық көбінесе қауіпсіздік сертификаттары арқылы немесе цифрлық қолтаңбаның басқа түрлері арқылы жүзеге асырылады.Біз оларды қатер мен шешімде жан-жақты қарастырамыз. Бастартпаушылық заңды міндеттеме болып саналады және екі тараптың келісімшарттарынан туындайды.Электронды саудада бұл тапсырыс берушінің бизнес және мәміле үшін төлеуге міндетті екендігін, қызмет көрсетуді орындауға немесе нысан мен қызметті ұсынуға міндетті екендігін білдірер еді.Бұл екі сала кәсіпорындарға ғана әсер ететіндіктен, ХҚЕС ядро принциптері болып танылмайды.

ХҚЕС бірінші кезекте АТ қорғаумен байланысты. IAS ұйымның ақпараттық қорын, осы активтердің осалдығын, активтерге зақым келтіретін қауіптерді,сондай ақ активтерді қорғау саласындағы саясатты анықтайтын тәжірибелер қатарын біріктіреді.Осы талдаудың соңғы нәтижесі болып, тәуекелдің стратегиялық талдауы ретінде белгілі,механизмге ақпараттық қауіпсіздікпен қамтамасыз ету үшін көшірілетін қауіпсіздік саясаты болып табылады. Біз назарымызды тәуекелді басқару процесіне шоғырландырамыз. Тәуекелді басқаруды орындау үшін ұйымның ең алдымен мақсаты болу керек.Осы мақсаттар тәуекелді басқарудың бірнеше кезеңінде пайдаланылады.Мақсаттар «ұйым қауымдастық мүшелеріне тұтастықпен және қайырымдылықпен жоғары сапалы қызмет ұсынады» деген сияқты, тек қана ең биік деңгейде мәлімделеді. Мақсаттар

басым бола алады, кейбір мақсаттар басқалармен қарама қайшылыққа түсе алады. Мысалы, клиенттерге қайырымды болу мақсаты деректерді тұтастықпен қамтамасыз ету мақсатымен қарама қайшылыққа түсе алады, тез ден қоюды талап етеді, соңғы жағдайда деректер пайдаланар алдында мұқият тексерілуі талап етеді.

Қазіргі кезде ұйым тәуекел бағасын орындауға міндетті. Бағалауда ұйымның ақпараттық активін анықтау бірінші қадам болып табылады. Бұл активтер (мысалы компьютерлер, компьютер желілері, адамдар), интеллектуалды меншік (идеялар, өнімдер), сондай ақ ақпарат (концентрацияланған және өңделген деректер). Активтерді анықтаудан басқа олар түрлеріне қарай жіктелуі мүмкін (мысалы аппараттық қамтамасыз ету, процесс, персонал) және олардың ұйым үшін маңызды басымдығы. Жіктеу, ақпарат қауіпсіздігі жағдайында кейде қолданылады. Бизнес «қоғам», «сезімтал», «жеке», «құпия» сияқты санаттарды өздерінің клиенттеріне жинаған деректерді сипаттау үшін қолдана алады. Қоғамдық ақпарат аттар мен мекенжайларды қамти алады (себебі, бұл ақпарат телефон кітабы арқылы қол жетімді). Сезімтал ақпарат телефон нөмірлерін және электронды поштаның мекенжайын қамти алады. Бұл ақпарат көпшілікке арналмаса да, бұл ақпарат, егер ERS бұл туралы білуге міндетті болса, адамның жеке өміріне қол сұғушылық қаупі бар деп қарастырылмайды. Жеке ақпарат ол әлеуметтік қамсыздандыру, несие картасының нөмірі немесе денсаулық, білім, ақпараттар үшінші бір тұлғаларға беріліп жатса, қатер төндіре алатын ақпарат болып табылады. Бұл ақпарат жария болудан федералды заңнамамен жиі қорғалады. Соңында, құпия ақпарат ұйым патенттелген ақпараттан және бизнес-жоспар сияқты ақпараттарды құпия ұстайды деген ақпараттан тұрады. Үкімет құпия және аса құпия сияқты жіктелген тегтер мен осы төрт санаттың шеңберінен шығып кетеді.

Тәуекелді бағалауда келесі қадам болып әрбір активтің осалдықтарын анықтау болып табылады. Осалдық активтердің арасында өзгеріп тұрады, бірақ көптеген активтер, олардың санаттылық түріне негізделген, осалдықты бөлетін болады. Мысалы, елестетіп көріңізші, тәуекелді бағалауда активтердің үш түрі менеджерлер, техникалық персонал, сондай-ақ әкімшілік персонал болып табылады. Бұл адамдардың барлығы, және кейбіреуі олардың осалдығынан бірдей болады. Кез келген тұлға басқа ұйымға, неғұрлым жоғары жалақы үшін ауып кетуі мүмкін, және кез келген тұлға әлеуметтік инженерияның шабуылына осал болуы мүмкін (қамтылатын қатерлер мен шешімдер). Осылайша, көптеген аппараттық элементтер электр қуатын ажырату және кернеудің ауытқулары сияқты белгілі бір осалдықты бөліседі, алайда аппараттық

құрылғылардың бәрі рұқсатсыз енудің осалдығын қамтымайды, себебі кейбір аппараттық құралдардың желіге біріктірілу мүмкіндігі жоқ. Осалдық анықталғаннан кейін, тәуекелдерді бағалау қатерлермен жалғасады. Яғни, осалдықты ескере отырып, зиянкестердің осалдықты пайдалана алатын қандай ықтимал тәсілдері болуы мүмкін? Алдыңғы абзацта көрсетілгендей, адам әлеуметтік инженерлік шабуылдарға осал болуы мүмкін.

Дегенмен, құпия файлдарға қол жеткізу мүмкіндігі бар біреуге деген мұндай шабуыл жасау мұндай мүмкіндігі жоқ адамға шабуыл жасағаннан көп қаупі бар.

Осылайша, құпия деректерді сақтау файл серверіне рұқсатсыз қол жеткізу тек презентациялық графика үшін қолданылатын принтер немесе портативті компьютерге қол жеткізуге қарағанда әлдеқайда үлкен қауіп болып табылады.

Тәуекелді бағалауды аяқтадық, ұйым әрбір тәуекелді өңдеуді анықтай отырып алға жылжиды. Тәуекел, мақсат сияқты, басым болуға міндетті. Мысалы, ұйымдастырушылық мақсаттардың бірі «ең жақсы және талантты қызметкерлерді жалдау» болуы мүмкін, бірақ «қызмет көрсетудің жоғары сапасы» мақсаты үлкен мағынаға ие. Осылайша, «қызмет» мақсатын қатерге тігетін тәуекелдер «жұмысшы» мақсатқа қатер төндіріп тұрған тәуекелге қарағанда аса үлкен басымдыққа ие. Тәуекелдер басымдығында тәуекелді басқару кейбір тәуекелдер қолайлы болып табылатындығын анықтай алады. Бұл, мысалы, бұл тәуекел қызметкерлердің басқа ұйымдарға ауып кету тәуекелділігі болуы мүмкін. Дегенмен, веб-порталға шабуыл жасау сияқты (мысалы, қызмет көрсетуден бас тарту) ұйымдарға сыни зақым келтіретін, біраз уақыт аралығында бизнестің жоғалуына әкеліп соқтыратын кейбір тәуекелдер сәйкестендірілуі мүмкін. Тәуекелдер, өз ретінде тәуекелдің азаюына әкелетін саясаттың құрылуын талап етеді. Мұндай саясат болуы, содан кейін осындай саясат қабылдануы тиіс. Саясат адами ресурстар саласындағы тәжірибені жалдау, орта буынды басқару үшін басқару әдістері, сондай ақ АТ қызметкерлерінің қолданған техникалық шешімдері сияқты нақты ұйымдастыру саласын қамтуы мүмкін. Тәуекелді басқару жоспары жүзеге асқаннан соң, ол бақыланып отыруға міндетті. Мысалы 6 ай өткен соң әртүрлі персонал нәтижесін тексеруі мүмкін. Осындай санкциялар немесе өздігінен білім алуды ұйымдастыру тиісті түрде жұмыс істемейтін жоспар саласын анықтауға әкелуі мүмкін. Осылайша, тәуекелдерді басқару процесі итеративті және тұрақты негізде болып табылады. Емтихан арасындағы уақыт бірнеше өлшемдерге негізделетін болады. Бастапқыда тәуекелдерді басқару жоспары жақсы жұмыс істе-

генше итерация бірнеше айға созылуы мүмкін. Итерация кейін жаңа активтердің жиі туындауына, осалдықтар мен қауіп-қатерлерге байланысты айдан бірнеше жылға дейін созылуы мүмкін. Тәуекелдерді басқару жоспары жауаптардың кең спектрін қамтуы мүмкін. Жауаптар жаңа рәсімдер жасау үшін АТ-операцияларының іс-қимылдарына байланысты әр түрлі болуы мүмкін. Жеке әрекеттер мысалы, камералар, қадағалар мен өрт сөндіру дабылдамасын, компьютердің бұғаттау тетігін орнату және монитор немесе қарауылға қосымша қызметкерлер жалдауды қамтуы мүмкін. Желіаралық экрандар сияқты АТ операциялары келесі бөлімде талқыланады. Жаңа рәсімдер жалдау процестерін, АТ қолдану саласындағы саясат және қолжетімділікті бақылау, ақпаратты басқаруды, активтер және техникалық қызмет көрсетуді және жанарту кестесін қамтуы мүмкін.

Тәуекелдерді басқарудың тағы бір компоненті болып авариялық қалпына келтіру жоспары болып табылады. Авариялық қалпына келтіру жоспары апатқа жауап ретіндегі іс-қимылдар жоспарын функционалдығын жоғалтып алу дәрежесіне дейін әсер ететіндей етіп ұйымдастыру. Табиғи апаттардың бірнеше түрі бар, сондай-ақ қалпына келтіру жоспары барлық (немесе бірнеше) табиғи апаттар түрлерін қамтуы мүмкін, немесе табиғи апаттардың әрбір түрі үшін бірегей қалпына келтіру жоспары бар болуы мүмкін. Апат катал ауа-райына негізделген, құйын, дауыл, жер сілкінісі мен су тасқыны сияқты табиғи болуы мүмкін. Табиғи апаттар техногендік болуы мүмкін, бірақ кездейсоқ, өрт сияқты, инфрақұрылымның коллапсы (мысалы, ғимараттағы төмен түсіп кеткен еден), бірнеше күнге созылған электр қуатының өшірілуі немесе химиялық төгілу. Апат сондай-ақ, антропогендік бірақ мақсатты болуы мүмкін, террорлық әрекет немесе соғыс әрекеті сияқты. Төменде соңғы екі санаттағы мысалдар келтірілген. 2008 жылы шығыс жағалауындағы электр қуатын беруді тоқтату кездейсоқ деп танылған, бірақ электр қуатын өшіру кейбір қауымдар үшін бірнеше күнге созылды. Және, әрине, 2001 жылдың 11 қыркүйегінде лаңкестер мінген төрт ұшақ Нью-Йоркте, Вашингтонда, Колумбия округі, және Пенсильванияда жарылды, ол үлкен түсінбестік тудырады және әуе компаниясының бірнеше күн өшіп тұруына әкеледі. Авариялық қалпына келтіру жоспары үш нәрсеге көңіл аударуы керек. Біріншіден, алдын алу шаралары апаттан құтылуға әрекет еткенде қажет. Мысалы, өрт дабылы мен спринклерлі жүйе өрт таралуының, АТ және инфрақұрылым зақымдануының алдын алуға көмектеседі. Тиісті рәсімдер химиялық заттардың ағып кетуі кезінде қауіпсіз жерге көшіруді, соның ішінде барлық АТ-ресурстарды тез жабу немесе АТ-өңдеуді басқа сайтқа жіберу мүмкіндігін қамтуы мүмкін.

Жай профилактикалық шаралар үздіксіз қорек көздерін және жүйелік детекторларды компьютерлер тиісінше жабық болуына кепіл болуды, компьютерге зиян келтіретін электр кернеуінің ауытқуларынан қорғауды қамтуы тиіс. Екіншіден, апат жағдайын байқау үшін кейбір ой болуы тиіс. Әлбетте, қатты құйын немесе дауылды байқауға болады, бірақ кернеудің ауытқуын оңай байқау мүмкін емес. Және, ақыр соңында, авариялық қалпына келтіру жоспары апаттың өзін қалай жеңіп шығу керектігін шешуі тиіс. АТ назар аудару оларды апатына жақсы дайындалуға және өңдеуге өте қолайлы. Біріншіден, барлық деректердің тұрақты және уақытылы резервтік көшірмелер жасалуы тиіс. Бұдан басқа, деректердің сақтық көшірмесі алаңнан тыс жүргізілуі тиіс. Молшылықты дискті қоймасындабарынша мүмкін дәрежеде пайдалану қажет. Бұл мәселе келесі бөлімде толығырақ талқыланады. Әрқашан үздіксіз қоректендіру және жабдықтар жағындағы желілік сүзгілер сияқты қарапайым шараларды сондай-ақ күшті саясаткер құпиясөздерді және вирусқа қарсы бағдарламалық қамтамасыз етуді пайдалану керек. Жабдықтар мониторингісі (мысалы, камералар бар) және жабдықтарға қол жеткізетін, құпия деректерді сақтайтын және сондай-ақ, іс-әрекеттерімен пайдалы персоналды тіркеу.

Және, әрине, өрт дабылы түріндегі отқа төзімді, өрт сөндіргіштерді, эвакуациялау жоспарларын, , тіпті өрт сөндіру көбігі бәрі де мүмкін.

Дайындау авариялық қалпына келтіру жоспарының тек бір бөлігі. Қалпына келтіру басқа бөлігі. Әрекеттер еркін АТ-жабдықтары бар (бүлінген немесе жойылған жабдықтарды тез қалпына келтіру мүмкіндігі) және деректердің сақтық көшірмесі қызметі бар, тек кейбір ықтимал нұсқасы ғана баруға дайын қалпына келтіруді қамтиды. Көптеген ірі ұйымдарда, АТ бір сайттан астам бөлінген, сондықтан бір учаскедегі апат ұйымның жұмысына зиян келтірмейді. Табиғи апат кезінде басқа сайт (тар) жүкті алып кетеді. Қалпына келтіру бастапқы сайт сақтық көшіруді және деректер мен прцестерді кері осы сайтқа қайтарылуын талап етеді. Тәуекелдерді басқару жоспары жиі басшылықпен бірге алынған. Бірақ бұл прцеске қосу үшін, АТ-персоналға сүйенген, кез келген ұйым үшін өте маңызды болып табылады. Бұл, ең алдымен, басқарма технологиялардың барлық осалдықтары мен қауіп-қатерлерін білмейді. Сонымен қатар, жаңа қауіп-қатерлердің жиі пайда болуына қарай уақтылы үлес алу үшін бұл маңызды болып табылады. Сізді мансабыңыз кезінде тәуекелдерді басқаруға тартып жатса таң қалмаңыз.

Мәселелер мен шешімдері

Ақпараттық қауіпсіздік өз негізінде ұйымның деректерін / ақпараттарды қорғауға тиіс. Дегенмен, қоршаған деректердің бірнеше қабат бар. Бұл операциялық жүйе, бағдарламалық қосымшалар, компьютер, оның ресурстары, желі және пайдаланушылар. Осы қабаттардың әрқайсысында әртүрлі қауіп-қатерлер болады, және олардың әрқайсысында өзінің қауіпсіздік нысаны болады.

Кейбір қауіп-қатерлер әр түрлі деңгейде болып жатқандықтан, біз қабаттарды емес қауіптерді қарастыратын боламыз. Дегенмен, ақпараттық қауіпсіздік әрбір қабатты қорғамайынша, толық болмайды. Әлеуметтік инженерия пайдаланушыларға арналған қауіпті білдіреді.

Идеясы мынау: қолданушы ол жиі және өте оңай алдануы мүмкін, сондықтан қолданушы әлсіз буыны болып табылады. Әлеуметтік, инженерлік шабуылдың қарапайым мысалы, бұл қалай жұмыс істейді. Сіз бір кеште телефон қоңырауын қабылдадыңыз. Дауыс өзін АТ деп сйкестендіреді, және жұмыс серверінің ауытқушылықтарына байланысты өзінің аккаунтын қалпына келтіру үшін құпия сөзге мұқтаж екендігін айтады. Құпия сөзсіз сіздің барлық деректеріңіз жоғалуы мүмкін. Сіз оларға өз құпиясөзіңізді айтасыз. Енді олар енді сіздің есепшотыңызға ене алады, өйткені олар АТ емес, бірақ шын мәнінде ол зұлым ниетпен.

Әлеуметтік инженерия адамдардың құпиясөздерін, банктік шоттарының нөмірлерін, несиелік карта нөмірін, әлеуметтік сақтандыру нөмірін, PIN (жеке идентификациялық - BER), және басқа құпия ақпаратты алуға пайдаланылады. Әлеуметтік инженерия телефон қоңырауына қарағанда әлдеқайда жұқа болуы мүмкін. Әлеуметтік ортада, ақылды хакер содан кейін құпиясөзді бұзуға пайдалануы үшін сіздің ақпаратты жария етуіңіз әлдеқайда мүмкін. Мысалы, сіздің үй мысықтарын жақсы көретініңізді біле тұра, біреулер мысықтарыңыздың атын, біреуінің атын құпиясөз ретінде пайдаланатыныңызды көргісі келіп, білуге талпынуы мүмкін. Көптеген адамдар өзінің сүйікті атын кейіннен санмен құпиясөз ретінде енгізу үшін пайдаланады. Әлеуметтік инженерияда вариация қолданушыларды алдау үшін ақпаратты электрондық түрде бұрмалайды. Фишинг қандай да бір операцияны орындау, үшін оларды веб-сайтқа бағыттау үшін, адамдарға арналған электрондық поштаны қамтиды. Веб-сайт, алайда, бұл көрініп тұрғандай емес. Мысалы, біреулер веб-сайтты кредиттік карталар сайтына ұқсас болу үшін құрып алады. Енді кейбір кредиттік карточка компания клиенттеріне электронды пошта арқылы олардың өз тіркелген жазбаларына кіру керектігі немесе шоты жабылатындығы туралы хабарланады. Сілтеме электронды по-

чта арқылы жасалынған, алайда, оларды спам-сайтқа жібереді. Қолданушы сілтемесін басады және жалған веб-сайтқа түседі. Онда, қолданушы ақпаратты енгізеді (қауіпсіз құпиясөз, кредиттік картаның нөмірі және т. б.), бірақ, бұл ақпараттың енді дұрыс емес адамға қолжетімді болатынын білмей. Компьютерлік жүйеге шабуылдайтын басқа класс қауіпі желідегі шабуылға өзі бағытталған, қолданбалы бағдарламалық қамтамасыз ету немесе операциялық жүйесі. Бұл класс хаттаманың шабуылын, бағдарламалық құралдарды, басып кірудің эксплойттарын және ішкі шабуылдарды қамтиды. Шабуылда хаттаманың әлсіздігі немесе кемшілігін қанай отырып бірі компьютер жүйесіне қол жеткізуге тырысады. Мысалы, белгілі қауіпсіздік мәселелері TCP / IP бар (басқару берумен / желіаралық хаттамасы). Жақындаудың бірі TCP айдап әкету деп аталады, онда қаскүнем хост-компьютердің IP-мекен-жайын пайдалана отырып хост-компьютерді желідегі жұмысын жасанды етіп шығарады. Хаттаманың көптеген шабуыл нысандары компьютерлік желі туралы ақпарат алу мақсатында барлау түрінде, нақты шабуыл алдындағы дайындық ретінде пайдаланылады. (Хаттама Internet Control Message) ICMP шабуылы әртүрлі хосттың компьютерлік желідегі IP-мекенжайларын білу үшін пинг бағдарламасын пайдалануы мүмкін. Қаскүнем IP-мекенжайды ашқаннан кейін, шабуылдың басқа түрлері іске қосылуы мүмкін. Шабуыл, шабуыл түрі, IP-спуфинг және ICMP (пинг) біріктіреді басқа құрылғының IP-мекенжайын бұрмалағандағы шабуыл желінің сірә, бір бөлігі болып табылады. Осылайша, қаскүнем кейбір ICMP қалыпты шабуылын жеңе алатын қауіпсіздік тетіктерін айналып өтуі мүмкін.

Бағдарламалық қамтамасыз ету эксплойттары сөз болып отырған бағдарламалық қамтамасыз етуге байланысты өзгеріп тұрады. Эксплойттардың SQL инъекциясы мен буфердің өте танымал екі түрлері. SQL инъекциясында қаскүнем SQL командасын URL-бір бөлігі ретінде веб-серверде береді. Веб-сервер, URL мекенжайының бір бөлігі ретінде, SQL командасын күтпей-ақ, сұрау қабылдай алады. URL сұрау "?" артынан келеді, Және өріс пен мәнді қамтиды, мысалы, www.mysite.com/products.php?productid=1. Бұл жағдайда, бет [products.php](http://www.mysite.com/products.php) веб, ең алдымен, деректер қорына `productid=1` алу үшін жүгінеді. SQL Web артынан осы деректер қорында жұмыс істеу үшін сұраныс орындалуы мүмкін. Мысалы, түрленген URL www.mysite.com/products.php?product=1; DROP TABLE oimder команда беруге болады SQL командасын `DROP TABLE oimder` өнімдерін деректер қорынан жоюға болар еді. Егер веб-серверлерден қорғалған және SQL командасын дерекқорға бере алса. Бұл SQL командасы деректер қорына қорғалған жазбалардан жою

жазбаларын өзгерту үшін бірдеңе жасауы ықтимал. Буфердің толып қалуы бәлкім, бағдарламалық қамтамасыз етудің және эксплуаттардың жақсы белгілі ең ескі нысандарының бірі, сондықтан бағдарламалық әзірлеушілер бағдарламалық қамтамасыз етуді жазғанда олардан қорғануға жай-күйі болуы тиіс. Алайда, бұл әрдайым олай емес, және бағдарламалық қамтамасыз етудің көптеген бөлігі әлі де бұл шабуылға бейім келеді. Буфер-бұл жай ғана айнымалы (әдетте массив), ол ақпараттық материалдар топтамасының мәндерін сақтайды. Буфер мөлшері шектелуі. Егер де бағдарламалық қамтамасыз ету буферге кірістіру шектелуі екендігіне кепілдік бермесе, онда буферге массивтен кейін жады орындары жеткілікті болатындай етіп қоюға болады, сондай-ақ. Жады деректер мен кодты сақтайтын болғандықтан буферді зиян келтіретін кодпен толтыруға болатын еді.

Ақ қалпақ Versus қара қалпақта

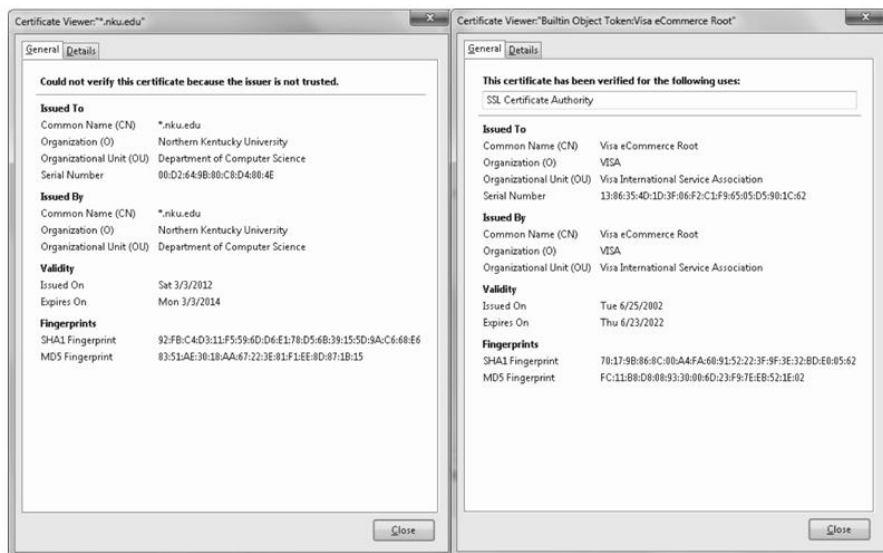
Хакер терминінің деп үш түрлі маңызы бар:

- кодты бұзатын біреу, яғни бағдарламашы
- компьютер хобби
- біреу компьютерлік жүйеге кіруге тырысады

Тарихи түрде қалыптасқан, хакер – ол бағдарламалық қамтамасыз етуді наразылық, құралы ретінде жасайтын адам. Мысалы, ерте хакерлер компьютерлік жұмыс істейтін телефон жүйелеріне тегін қалааралық қоңыраулар орналастыру үшін жиі бұзып кіретін. Оларды кейде фрикерлер деп атайды. Аса дәстүрлі хакерлерді, бағдарламашыларды, компьютерлік жүйелерге бұзып кіруге тырысатын адамды пайдалана отырып ажырату үшін бұзып кіруші деген термин ойлап табылды. Бұзып кіруші компьютерлік қауіпсіздікті бұзуға тырысады, бірақ тіпті осы анықтамамен де біз зиянды мақсатта жасайтын адам мен кемшіліктерді табу мақсатында, осыған орай қауіпсіздікті жақсартуға мүмкіндігі бар адамдар арасындағы айырмашылықты жүргізе білуіміз керек. Бірінші жағдай қара қалпақтағы хакер (немесе жай ғана қара қалпақта) деп атайды. мұндай адам террор немесе қылмыс жасау мақсатында қауіпсіздікті бұзады. Соңғы жағдайда жекелеген қауіпсіздік жүйелерін қара ниетсіз ажырататын деп ақ қалпақты хакер деп атайды (немесе жай ғана ақ қалпақта). Дегенмен, тіпті «ақ қалпақ» үшін де компьютерлік қауіпсіздік бұзу бұрынғыдай әдепсіздік.

Нүкте жағдайында, өздерін аноним деп атайтын, крекерлерді ұйымдастыру әр түрлі ұйымдардың наразылық, құралы ретінде компьютерлік жүйелерін бұзушылық болып табылады. Олар балаларға қатыгездік көрсетті деген арызына жауап ретінде католик шіркеуінің жоқтығына қарсылық белгісі ретінде Ватиканға шабуыл жасады. Олар әділет Де-

партаментіне қарасты сайттарға шабуыл жасап, Американдық дыбыс жазу компаниясы ассоциациясы, және Американдық кинокомпания ассоциациясы қарсылық ретінде антипираттық заң шығарушы ұйымдары Wikileaks активтерінің бағасын ұстап тұрғанда PayPal, MasterCard және Visa сайттарына шабуыл жасаған.



Алайда бұл шабуылдар қарсылық түрі деп жарнамаланса да, оларды көптеген халықаралық заңдарды бұзған сақтық операциясы деп қарастыруға болады.

Жадыда сақталады, процессор бұл код орындай алады, осылайша, зиянкес қойған операциялар орындауы мүмкін.

Ену және басқа да белсенді шабуылдар нысандары әдетте компьютерлік жүйеге рұқсатсыз кірудің қолжетімділігін арттыра отырып біріншінің маңайында айналады. Кіру үшін зиянкес операциялық жүйенің немесе желінің қауіпсіздік жүйесінде осал жер табуы керек немесе басқа біреудің тіркелгісімен қол жеткізе алады. Бұл үшін зиянкес пайдаланушының есептік жазбаның аты мен құпия сөзін білуі керек. Жоғарыда айтылғандай, әлеуметтік инженерия мен фишингте құпиясөздер алу әдістері бар. Құпиясөзді алудың басқа да тәсілдері қолданушының есептік жазбасына үнемі кіруге, сөздіктің әрбір сөзін білуге тырысатын бағдарламаларды жазуды қамтиды. Екінші тәсіл құпиясөзді білу үшін жай ғана адамды қадағалау, бәлкім, оны қадағалай отырып оны енгізуге

болады. Немесе, егер адам құпиясөзді төменде жазса, оның офисіне қол жеткізе алсаңыз, онда сіз оны енгізу үшін өз компьютеріңіздің айнала-сынан қарай аласыз. Тіпті пост жазылған құпиясөзді табуға болады - бұл мәлімет, компьютердің мониториянда тұрып қалған!

Құпиясөзді алудың тағы бір құралы пакеттер талдауы арқылы алу болып табылады.

Мұнда қаскүнем компьютерден тзүсетін хабар трафигін зерттейді, мысалы, электрондық пошта. Қолданушы электрондық пошта арқылы оған немесе өзіне құпиясөзді жіберуі әбден мүмкін (біз жіберді деп үміттенеміз, бұл екіталай). Адамдардың құпиясөзін табудан бөтен, операциялық жүйеге кіру үшін пайдаланылатын операциялық жүйенің басқа да осал тұстары бар. Unix операциялық жүйесі біреуге Unix жүйесіне құпиясөзді мүлдем пайдаланбай телнет жүйесі бар қарапайым кемшіліктері болды. Ішіне кіріп алған бұзушы шабуылды ашып жіберуге болады. Белсенді шабуыл файлдардың жойылуынан, мәліметтерді көшіру, сондай-ақ файл деректерін өзгерту, артынан қандай да бір зиянды кодты қалдыруы немесе кері - есіктер тіркелгісін жасауы мүмкін (зиянкеске кез келген уақытта кіруге мүмкіндік береді). АТ-жүйесі қауіпсіздігі серпілісіне, ішкі жұмыс арқылы одан да қарапайым тәсілі. Егер сіз қолжетімділігі бар адамды білсеңіз, олардың кез келгенін зиянкес сатып алуы мүмкін, олай болса зиянкес файлдарды өшіріп тастауы, көшіруі өзгертуі, оның дұрыс емес ыдыс-аяққа кірістіруі, немесе сол адам арқылы басқа жолмен компьютерлік жүйенің сипаты туралы біліп алуы мүмкін.Бұл компьютерлік жүйедегі ең осал бөлшегі,себебі адамдар сенім білдіре отырып жартылай қолжетімділік алады.Бұл сенімділік бұзылса, басқа енудің түрлеріне қарағанда, онда ол елеулі мәселелер тудыруы мүмкін.Жеке қолданушыларға жасалған шабуыл зиянды бағдарламалар шабуылының ең нашар түрі болып саналады.Зиянды бағдарламалар түрінің түпнұсқасы троян аты деп аталады. Троян аты бағдарламалық қамтамасыз етудің бір бөлігін ынталандырады, бірақ бұл шын мәнінде олай емес. Елестетіп көріңізші, сіз өте тиімді қосымша жүктедім деп ойлайсыз.Дегенмен, бағдарламалық қамтамасыз ету іс жүзінде қосымша файлдық жүйесі зиянды операциялар орындайтын кейіп танытады. Троян аты компьютерлік вирус бір түрі . Негізгі айырмашылықтары мынада: вирус басқасының ішінде тығулы,өзін көшіру қабілеті сондай, ол дискета арқылы (бұрын, біз оларды пайдаланған), флэш-диск, немесе қосымшалар, электрондық пошта арқылы.бір компьютерден екінші компьютерге көшіре алады. Дегенмен зиянды бағдарламалардың басқа да нысандары желілік құрттар болып табылады, олар компьютерлік желілерге және шпион-бағдарламаларға шабуыл жасай-

ды, (мысалы, Моррис Интернет-құрт талқыланады 12-тарауда). Белгісіз қолданушылардың веб-сайттарына қол жетімді болғанда Spyware жиі жүктеледі. Тыңшылық БҚ сіздің жүріс-тұрысыңызды бақылап тұруы мүмкін немесе есеп сайтына құпия ақпаратты, мысалы, сіз веб-нысанға енгізген кредиттік карта нөмірін беруі мүмкін. Сіздің бағдарламалық қамтамасыз етуіңіздің кейбіреуін зиянды бағдарламаның тағы да бір түрі жаулап алады. Мысалы, ол DNS ақпаратты басқа DNS өту үшін бағыттауы мүмкін, сұрау салуларға оң IP-мекен-жайымен қамтамасыз етудің орнына, әрқашан сіздің веб-браузеріңіз дұрыс емес орында (орындарда) қабылдайтын, жалған мекен-жаймен қамтамасыз етеді. Бір түпкілікті ортақ болып табылатын шабуыл нысаны, бүгін, атап айтқанда, веб-сайттарға, қызмет көрсетуден бас тарту болып табылады. Бұл қызмет көрсетуден бас тарту шабуылында, бір немесе бірнеше қарақшылық шабуыл жасағандар серверді соншалықты көп кіріс хабарлармен басып тастауға тырысады, ал сервер қалыпты бизнесте өңдеуге қабілетті емес. Қызмет көрсетуден бас тарту шабуылын орындаудың ең қарапайым тәсілдерінің бірі мыңдаған немесе миллиондаған немесе одан да көп) HTTP-сұраныстар жіберу. Дегенмен, ол тек трафикті арттырады; ұзақ уақыт бойы барлық сұрау салуларға сервердің жауап беруіне шектеу қою міндетті емес. UDP шабуылы UDP пакетінің (12-тарауда белгіленген) ішіндегісін зиянкес салып қойған басқа нәрсемен ауыстыра алады. Жаңа мазмұны сервердің кейбір еңбекті көп қажет ететін операцияларды орындағанын талап етуі мүмкін. UDP су тасқынын орындай отырып үлкен серверлер жабық болуы мүмкін. Қызмет көрсетуден бас тарту шабуылының басқа да нысандары TCP SYN және шабуыл толып TCP ACK ретінде белгілі TCP / IP хаттама хендшейкингті пайдаланады. Жоғарыда келтірілген ой-пікірлер сыналған шабуылдар түрінің толық тізімі емес. Және, әрине, жана шабуылдар типтері әр жылда жаңарады. Ақпараттық қауіпсіздікке, қорғау тетіктеріне жәрдем беру үшін, осы қауіп-қатерді қолайлы тәуекелдерге дейін шектеу үшін бізге не керек. Біріншіден, осы ұйымның пайдаланушылары білімді болуы тиіс. Әлеуметтік фишинг және тыңшылық нысандарын біле отырып пайдаланушылар өз құпиясөздерін қалай қорғау керектігін біле алады. Сонымен қатар, пайдаланушылар тек сенімді парольдер қолдана талатынына саясат кепілдік беруге тиіс, сондай ақ өз құпия сөздерін жиі ауыстыра алады. Кейбір жағдайларда «сіз білетін» деп аталатын парольдің орнына ұйым басқа тәсілді қолданады. Оның орнына басқа екі тәсілді "сіз" және "сен кімсің". Бірінші жағдайда қол жеткізу процессі қандай да бір кілтті иеленуді қамтиды. Неғұрлым кең тараған нысаны кілт болып табылады кілт-карта (магнитті карта). Мүмкін, бұл құпиясөзбен бірге пайдаланылуы мүмкін,

сондықтан сіз кілтке ие болуыңыз және білуге жүйеге кіруге арналған парольді білуіңіз керек. Соңғы жағдайда, "сен кімсің" қандай да бір айтылмайтын физикалық аспекті білдіреді. Бұл жерде биометрия саусақ ізі болсын, матчтың дауысты сәйкестендіру болсын немесе тіпті ерін баспасы болсын, метриканың қайталануы мүмкін емес. Бұдан әрі, пайдаланушының қол деткізе алатын ресурстарға қол жеткізуін қамтамасыз ету. Бұл кіру бақылауын қамтиды. АТ-саясатының бөлімі немесе тәуекелдерді жоспарлау басқармасы пайдаланушы қолжетімділік алатын механизм тетіктерін қамтиды. Бұл құқықтар пайдаланушы қол жеткізетін файлдарды қамтуы тиіс, қолданушы қол жеткізе алмайтын кез-келген файлдарды қамтымайды. Қол жеткізу құқықтары әдетте операциялық жүйеде қандай да бір қолжетімділікті басқару тізімі тетігінің көмегімен орындалады (қараңыз: 6-тарау осы мәселе бойынша талқылау үшін). Windows және Linux операциялық жүйесінде әрбір пайдаланушы үшін, не (Windows, Linux Security Enhanced) немесе 9-биттік gwxgwxgwx схемасы (Linux) тізімдері пайдалана отырып, бақылау қолжетімділігіне, файлдар мен каталогтар шектеледі. Көптеген қазіргі заманғы ДББЖ қатынасты басқару рөлі негізінде пайдаланады. Осындай жүйедегі рөлдер анықталған, олар қол жеткізу әдістерінің тізімін қамтиды. Мысалы, басшы барлық деректерге, толық қол жеткізе алса, ал менеджерлер өз бөліміндегі деректерге ғана қол жеткізе алады. Талдаушы деректер тек оқу үшін берілуі мүмкін, белгілі бір қатынастарға қол жеткізу. Кейін рөлдер анықталғаннан кейін, индивидуумға тағайындалған рөлдер. Бір адамда ақпараттық қол жеткізу құқықтарына мүмкіндік беретін әр түрлі рөлдер болуы мүмкін. Мақсатты компьютерлік жүйенің басқа да көптеген нысандары. Қорғау басып кіру, шабуыл, қызмет көрсетуден бас тарту, бағдарламалық эксплойттар және зиянды бағдарламалардан, рұқсатсыз қол жеткізуден қорғайтын технологияларының үйлесімділігін қамтиды. Шешім белгілі бір хабарламаларды желіге кіру немесе шығу болыrmас үшін желіаралық экранды, вирусқа қарсы бағдарламалық қамтамасыз етуді, сондай ақ енулерді анықтау үшін бағдарламалық қамтамасыз етуді қамтиды.

Шешім қамтиды желіаралық экран болдырмау үшін белгілі бір түрлері хабарламаларды кіру немесе шығу желісі, анти-вирустық бағдарламалық қамтамасыз ету үшін іздеу зиянды бағдарлама БОЙЫНША, сондай-ақ бағдарламалық қамтамасыз ету үшін енулерді. Рұқсат бағдарламалық қамтамасыз ету эксплойттары талап етеді анықтау бірінші ерлігі, содан кейін түрлендіруді бастапқы кодты бағдарламалық қамтамасыз ету. Егер де бағдарламалық қамтамасыз ету болып табылады коммерциялық өнім немесе ашық бастапқы коды бар, әдетте, кейін қолданады

табылды, жаңарту шығады қолданамыз, оның ішінде бірнеше күннен бірнеше аптаға дейін. Шабуылдардың бас тарту қызмет көрсету және басып кіруді болып табылады неғұрлым күрделі міндеттері, ал АТ-персонал үнемі қорғай оларға қарсы.

Екі түрі қауіп туындайды қарым-қатынас кезінде қорғалған ақпаратты Интернет арқылы, бұл ма толтырып веб-форманы жіберу немесе ақпаратты электрондық пошта арқылы жіберіледі. Біріншіден, интернет-коммуникациялар бойынша өңделеді қарапайым мәтінмен. Егер біреу ұстап хабарламалар, ұсыныстар қорғалған ақпарат ашылған оқу үшін. Екіншіден, жөнелтуші тиіс деп сенемін алушы, кім олар, яғни, алушы тексеруден өтуі тиіс түпнұсқалығын. Шешу үшін бірінші міндеті, шифрлау пайдаланылуы мүмкін. Шешу үшін екінші бір мәселе, біз пайдалана алар еді үшінші тарапқа шынайылығын тексеру үшін алушы. Өңдеу үшін екі осы шешімдерді, біз көшеміз цифрлық қолтаңбалардың және сертификатов.

Идея сертификаты немесе цифрлық болып табылады, яғни қолданушы (веб-сайты) құрады белгісі, көрсету үшін, кім олар бар. Бұл жай ғана деректерді қамтуы мүмкін, мысалы, электрондық пошта қолы, сурет немесе аты. Бұдан әрі, осы орналастырылады бірге файл. Технологиясы шифрлеу (қарастырылуда, келесі бөлімде) пайдаланылады генерациялау үшін кілт. Кілт және деректер беріледі сертификаттау орталығы. Бұл компания, оның жұмысы, оның ішінде мемлекеттік кепілдік беру үшін, қолданушы, олар бекітеді, және орналастыру, оларды басып шығару мақұлдау деректер файлы. Бұл нәтиже қол қойылған сертификат немесе цифрлық қолтаңба. Қол қойылған құжат болып табылады бір мезгілде дәлел түпнұсқалығын және шифрлау үшін қажетті ақпарат жөнелтуші жіберу үшін қорғалған ақпаратты қикар алаңдаушылық осы ақпарат мүмкін перехвачен және оларға қол жеткізу.

Егер пайдаланушы сертификаты немесе қолы, және ол ескірген, қол қойылмаған, немесе кімнің мәліметтер сәйкес келмесе және мәлімделген жеке пайдаланушы, онда алушы (ескерту, бұл адам болуы мүмкін емес, кімге олар бекітеді. Сурет 15.4 көрсетеді екі сертификат. Бір сол жағында университетінің және самоподписан. Сіз байқамайды, жоғарғы бөлігінде осы сертификат "өтінішті тексере алмады бұл сертификат, өйткені эмитент болып табылады сенімді." Бұл көрсетеді, бұл сайт пайдаланады сертификаты жоқ алаңдатты болуы тиіс сертификат орган қол қою, оларды толықты - тіфісате. Кез-келген пайдаланушы, сертификат алады ескертілді-бұл бизнес осы сайтты қауіпті болуы мүмкін, өйткені олар авторизациялаудан өтпедіңіз. Бақытымызға орай, бұл сертификат үшін қолданылады емес, қорғалған ақпаратты алу, сондықтан қолда-

нушы болмайды айтарлықтай тәуекелге. Сертификаттау ката оң жағында фигуралар, Visa-мен қол қояды SSLCertificateAuthority, осылайша, сайт сенуге болады.

Ақпараттық қауіпсіздік қамтамасыз етуге тырысады құпиялылық, тұтастық және нақтылау ақпараттың болуы, ол сақталады, өңделеді және беріледі (желі). Тәсілдері, жоғарыда аталған, ең алдымен, бағытталған ақпарат беруді желісі бойынша және ақпаратты өңдеу. Біз пайдалануға болады басқа шешім тиісінше сақтауды қамтамасыз ету үшін: артықшылығы. Графикалық бейнелеу құралдарын ұсынады, сондықтан деректер қол жетімді болып, сақтау құрылғылары бүлінген немесе желіден тыс. Бар жалпы көзқарас жәрдемдесу молдығы. Бірінші арқылы резервтік көшірмелерін файлдық жүйесі. Екіншіден, есебінен артық сақтау, атап айтқанда, RAID.

Резервтік көшірмелерін қалануы ең алғаш компьютерлік жүйелер. Сақтық көшірмесі ғана болып табылады көшірмесімен мәліметтер жинау. Резервтік көшірмелерін сақтау үшін қолданылатын магниттік таспадағы. Резервтік таспалар сақталады, жекелеген үй-жайларда, кепілдік беру үшін, егер ештеңе бүлінген компьютер жүйесінің, таспа қорғалуы тиіс. Осылайша, бұл деректер оңай қол жетімді қалпына келтіру үшін қажет болған жағдайда. Бір қызығы, көптеген таспаларды үшін пайдаланылатын датасы қойылған деректерді сақтау кері 1950-ші жылдары бұрыннан деградировали, және деректер жоғалуы.

Дискеталар мен оптикалық дискілер, сайып келгенде, деген лентасын максаттары үшін резервтік көшіру. Дегенмен, төмен құны сақтау қатты дискіде бүгін, көптеген пайдаланушылар болады сақтық көшірмесін файлдық жүйенің сыртқы қатты диск немесе пайдалануға бұлт сақтау. Резервтік көшірмелері, не болуы мүмкін барлық файлдық жүйесі, немесе сол бөлімнің файлдық жүйесін өзгертілген сәттен бастап соңғы резервтік көшіру. Соңғы жағдайда, бұл ретінде белгілі инкрементного резервтік көшіру. Операциялық жүйенің бапталуы мүмкін орындау үшін, резервтік көшіру, автоматты түрде. Жоспар шеңберінде тәуекелдерді басқару талап етіледі саясаты резервтік көшіру.

RAID білдіреді артық массиві тәуелсіз диск (балама ретінде, "Мен", сондай-ақ тұру үшін арзан). Идея RAID тұрады және бірыңғай сақтау құрылғылары деректер (әдетте түрінде шкафы), құрамында бірнеше үйлестірілген жестких.

Дискілі файлдар

Дискілі файлдар әдеттегідей блоктарға бөлінген, ал блоктар өздігінше жолақтарға бөлінеді. Блок-диск одан соң жолақтардың өлшеміне байла-

нысты бір немесе бірнеше дискінің үстіне сақталады. Шағын жолақтар бір уақытта көптеген дискіні пайдаланған кезде, үлкен жолақ бірнеше дискіні бірдей уақытта пайдалана алады. Осылайша, жллықтың өлшемдері қалай тәуелсіз дискілер пайда болатынын анықтайды. Мысалы, егер блок-диск дискінің кез келген жоғарғы бөлігі арқылы таралатын болса, содан соң диск-блокқа қол жеткізу анағұрлым жылдам, себебі барлық дискілер бір мезгілде қолжетімді болады, осылайша әрбір үстіңгі бет сонда сақтаулы биттерге қол жеткізу үшін уақытты аз кетіреді басқа жағынан алып қарағанда, егер RAID кабинеттер айталық сегіз бөлшектенген жылдамдық дискілерін пайдаланатын және жолақтар бар жоғы екі диск арқылы таралатын болса, онда RAID шкафты бірдей уақытта дербес төрт пайдаланушы қолдана алады.

RAID дискіге қол жеткізу уақытын қысқартатын «тәуелсіз» аспектіге қарамастан, RAID пайдалануға арналған шынайы себеп «мерзімі біту» болып табылады. Бірнеше қолжетімді жинақтаушы қолда бар болғанда, мәліметтер көшіріліп басылуы мүмкін, сондықтан егер сектор дискісі бір дискіде нашар жүретін болса, ақпарат шкафтағы басқа жерде қайталануы мүмкін. Яғни, қолданыстан шыққан мәліметтер нашар секторлардың кесірінен жоғалған мәліметтерді қалпына келтіру үшін пайдаланыла алады. Мерзімі біту ережеге сәйкес үш форманың бірін алады. Біріншіден, бір дискілер жинағы басқа бір дискілер жинағы арқылы көшірілетін айна форматы бар. Мұндай жағдайда дискілер кеңістігінің жартысы мерзімінің бітуін қамтамсыз ету үшін пайдаланылады. Бұл мерзімі біткен ақпараттардың анағұрлым көп санын қамтамсыз еткенмен, ең қымбаты болып саналады, себебі сіз резерв ретіндегі қолданылатын жалпы дискілік кеңістіктің жартысын жоғалтасыз. Мерзімі бітуді қамтамсыз етуге арналған Хэмминг кодтары мен нақтылық биттерін (3 тарауда қарастырылады) пайдаланудың басқа екі жолы. Екі нақтылық битасы мен Хэмминг кодтары мерзімі біту сатыларын қамтамсыз етеді, сондықтан қалпына келтіру аздаған зақымданулармен іске асуы мүмкін, бірақ аса ірі зақымданумен емес. Мерзімі бітудің бұл формаларының басымдылығы мерзімі біту ақпаратына қажетті жадының көлемі дискілік кеңістіктің жоғарғы жағына қарағанда анағұрлым аз, бұл формаларды айналы еріткішке қарағанда арзанырақ етеді.

RAID технологиясы қазіргі уақытта 1987 жылдан бері әрекет етуде және ірі ұйымдар үшін бұл қалыпты әрекет болып табылады. Жекелеген қолданушылар RAID шкафтарға қаражат салуы екіталай, себебі олардың құны анағұрлым қымбатырақ. RAID өзі түрлі деңгейлерден тұрады. RAID әрбір деңгейі түрлі басымдықтарды, кемшіліктер мен шығындарды ұсынады. RAID деңгейлері нөмірлермен белгіленеді, онда саны көп

болғаны RAID анағұрлым жоғарғы деңгейі дегенді білдірмейді. RAID деңгейлеріне қысқаша көзқарас.

RAID 0 мерзімі бітуді қамтамасыз етпейді, тек қана бірнеше дискілердің және жолақтардың көмегімен тәуелсіз қолжетімділікті ғана қамтамасыз етеді. RAID-1, басқа қырынан алғанда, бұл айна дискілердің тең жартысы айна ретінде пайдаланылатын формат. Осылайша, RAID 0 және RAID 1 толықтай қарама-қайшы спектрлер мен мерзімі біту құны қосымшалар: RAID 0, файлдарға арналған дискілердің барлық кеңістігінде пайдаланады, ең арзаны болып табылады, бірақ мерзімі бітуді ұсынбайды. RAID-1, резерв үшін диск кеңістігінің жартысын пайдаланады, ең қымбаты болып табылады, бірақ ең көп мерзімі бітуін ұсынады. RAID 2 мерзімі бітуін қамтамасыз ету үшін Хэмминг кодтарын пайдаланады және мүмкіндігінше аз мөлшердегі жолақтарды қолданады. RAID 3 мерзімі бітуді қамтамасыз ету үшін нақтылық биталарын және мүмкін болатын жолақтардың ең аз өлшемдерін пайдаланады. Хэмминг кодтары ең күрделі және оларды анықтау үшін көп уақыт кетеді. Одан басқа, RAID-2 жетістікке жету үшін дискілердің көп болуын талап етеді. Осы себепті RAID-2 қолданылмайды.

RAID 4, RAID 3 сияқты, нақтылық биталарын пайдаланады. Бірақ, RAID 3 қарағанда RAID 4 өзгешелігі, жолақтардың үлкен өлшемдерін пайдаланады. RAID 4-те, барлық нақтылық биталары сол дискінің өзінде сақталады. Осыған байланысты, RAID 4 тәуелсіз қолжетімділікке жол бермейді. Мысалы, егер екі процесс диск салғышқа бірдей уақытта енгісі келсе, екі жол екі жеке жолақтармен екі түрлі дискіде болғанына қарамастан, нақтылық туралы ақпарат бар бір дискіге түскенде, екі ену жолы соқтығысады. Осы себепті RAID 4 пайдаланылмайды. RAID-5те RAID 4 сияқты, бірақ нақтылық ақпараты дискілер бойынша таратылады. Осылайша, егер блоктар мен олардың нақтылық ақпараттары түрлі дискілерде болды деп есептесек, екі процесс екі түрлі диск блоктарына дербес ене алады. RAID-6 RAID 5 сияқты, айырмашылығы нақтылық туралы ақпарат көшіріледі және қосымша резервтеу және тәуелсіздік үшін дискілер арасында таратылады.

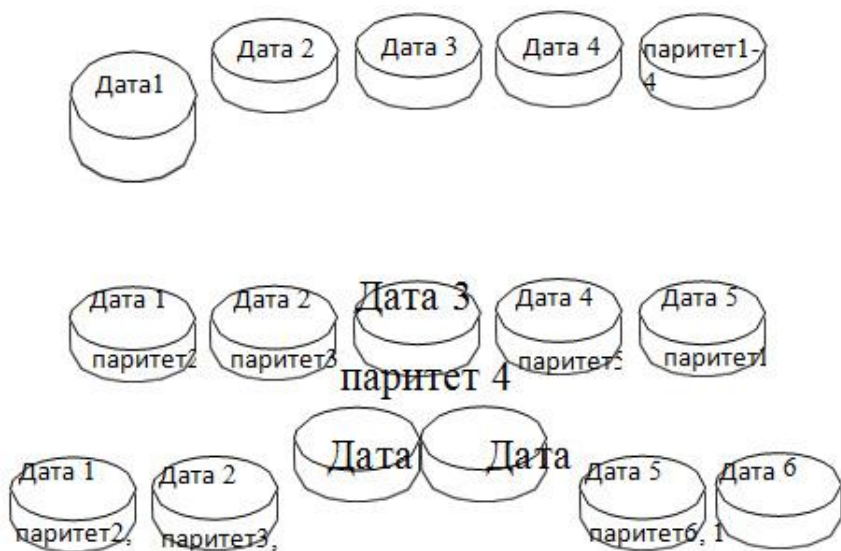
15.5 суретінде, RAID 4, 5 және 6 көрсетілген. RAID 4-те, нақтылықты бақылаудың барлық мәліметтері тек қана бір дискіде. Бұл диск тәуелсіз кірулер үшін тар орын болып табылады. RAID 5-те нақтылық мәліметтері барлық дискілер арасында бірдей таратылған, сондықтан бірнеше дискілік операциялар бір мезгілде жүргізілуі мүмкін. RAID 6-да нақтылық мәліметтері көшіріледі және таратылулар мерзімі бітуі ұлғайып қана қоймай, сонымен қатар бір мезгілде қолжетімді болуы ұлғаю мүмкіндігімен таратылады. Сендер RAID 6 кем дегенде бір қосымша

дискісі бар екенін және кез келген RAID 4 немесе RAID 5 қарағанда қымбат екенін байқайсындар, бірақ аса үлкен мерзімі бітуін, мүмкіндігінше жақсартылған қолжетімдікті қамтамасыз етеді.

Жақында ғана RAID 0 және 1 деңгейлерін үйлестірген RAID 0 + 1 немесе RAID 10 белгісімен белгілі гибридік деңгейлері шықты, сондай-ақ 3 пен 5 біріктіретін RAID 53 деп аталады. Озервтік көшіру және/немесе RAID шкафтардың көмегімен біз өзіміздің файлдарымыздың мерзімі бітуін, сонымен қатар қолжетімдік пен қауіпсіз сақталуын қамтамасыз ете аламыз. Сонда да, бұл аталған мәселені шешуге көмектеспейді. Бұл талапты қолдау үшін біз өзіміздің шешіміміз үшін тағы бір технологияны қосқымыз келеді: шифрлау. Шифрлау арқылы біз мәліметтер құпия түрде жүргізілетініне кепілдік беріп қана қоймаймыз, оның ұрлануынан қорықпай біз оны Интернет арқылы ашық ортаға жібере аламыз. Шифрлау күрделі тақырып болғандықтан, біз оны келесі тарауда сөз етеміз.

Криптография

Криптография – ақпаратты кодқа орналастыру ғылымы, сондықтан сенімді түрде беріле алады. Шифрлау дегеніміз кодқа енетін хабарламаны қайта құру процесі, шифрды бұзу дегеніміз шифрланған хабарламаны қайтадан кодқа қайта құру процесі.



15.5 суреті RAID 4, 5 және 6 сақталатын нақтылық туралы мәліметтер. Шығыс хабарлама. Шифрлауға арналған кілт, шифрланған хабарламалар ұрланған жағдайда да түсініксіз болатынына көз жеткізу.

Кодтарды пайдалану ғасырлар бойы қолданылып келеді. Кейде классикалық шифрлау деп аталатын, өткен кезеңдегі шифрлау негізінен қаламмен және қағазбен орындалатын. Хабарлама кодпен жазылатын, шабарман арқылы жіберілетін, ал алушы оны хабарламаға айналдырады. Егер шабарман қолға түссе (немесе сенімсіз болса), онда хабарлама оқылмайды, себебі оның коды болған.

Қарапайым код жай ғана алфавиттің әріптерін басқа әріптерге алмастырады. Мысалы, бұру коды әрбір әріпті алфавиттің астында бірқатар қашыққа орналастырады. Айналдырған кезде +1, оның 'a' 'b' айналады, ал 'b' 'c' айналады, тағы с.с., және 'z' 'a' айналып келеді. Екінші Дүниежүзілік соғыс кезінде немістің және одақтас елдің ғалымдары басқа тараптың кодтарын бұзуға тырысып анықтағыш құрылғылар құрды. Бүгінгі күні сенімді шифрлерді заманауи компьютерлер жеңіл бұза алатын кез келген шифрлеудің басқа формаларындай пайдаланады.

Бүгінде пайдаланатын криптографияның екі негізгі формасы бар: симметриялық кілт және асимметриялық кілт. Шифрлеудің қос формасы да мына идеяға негізделген, хабарламаны шифрленген және шифрленбеген күйде, және шифрленген түрден кері шифрленбеген күйге манипуляциялайтын кілт, математикалық функция ба. Шифрлеудің симметриялық кілтіне дәл сол кілт шифрлеу және шифрді ашу үшін пайдаланылады. Шифрлеудің бұл формасының позициясы пайдалы, тек қана хабарлама (немесе файл) басқаларға арналмаса, немесе қабылдаушыға кілт берілуі мүмкін. Осылайша, мысалы сіздер шифрлеудің бұл формасын файлды, оның мазмұнын ешкім оқи, көре алмайтындай етіп сақтау үшін қолдана аласыздар. Сіздер кілтті басқалармен бөліспегендеріңіз дұрыс, себебі басқа қолданушылар бұл файлды ашып қарауына мүмкіндік береді.

Интернеттегі электронды коммерцияның жағдайын қарастырайық. Сіз өзіңіздің веб-браузеріңізден компанияның веб-серверіне несие картаңыздың номерін жібергіңіз келді, сол үшін олар сауда үшін төлем алулары мүмкін (сіз оның сандарын жібергіңіз келеді, бірақ сауда жасағыңыз келеді). Сіз бизнес жүргізетін компаниялар оларға сенім артқаныңызды қалайды, сондықтан олар несие картасын шифрлеу үшін карта оларға берілгенге дейін шифрлеу алгоритмін пайдаланады. Мұны жасау үшін сіздің веб-браузеріңіздің кілті болуы тиіс. Сондай-ақ, олар сіздің веб-браузеріңізді кілттің көмегімен қамтамасыз ететін болса, сіз де кілтті ала аласыз. Егер сізде кілт бар болса, олардың серверлеріндегі басқа хабаарламаларды дербес қолға түсіріп және шифрлерді бұзар едіңіз, осылайша

басқа клиенттердің несие карталарыны номерлерін алар едіңіз. Осылайша, электронды коммерция үшін біз шифрлеудің симметриялық кілтін пайдаланғымыз келмейді. Оның орнына біз шифрлеудің асимметриялық формасына көшеміз. Шифрлеудің бұл формасында кілт басты орын алады. Бұл жабық кілт. Енді, құпия кілтті пайдалана отырып, олар ашық кілт жасаулары мүмкін. Ашық кілт еркін түрде әлдекімге берілуі мүмкін. Көпшілік кілтті есепке ала отырып, сіз хабарламаны шифрлейсіз. Оның үстіне, сіз ашық кілтті хабарламаның шифрін ашу үшін қолдана алмайсыз. Ұйым келіп түскен хабарламаның шифрін ашу үшін жабық кілтті өзінде сақтайды. Біз әдетте шифрлеудің симметриялық кілті деп, ол тек бір ғана кілтті, жеке (немесе құпия) кілтті пайдаланатындықтан позициялардың шифрлеудің құпия кілті деп айтамыз. Асимметриялық кілтті шифрлеуді көбінесе ашық кілтпен шифрлеу деп атайды, екі кілттің көмегімен, жабық кілт және ашық кілт. Шифрлеудің бұл екі формасы да 15.6 суретте көрсетілген.

Ашық және жабық кілттердің қос шифрленуі блокты көрсетуде пайдаланылады. Блок тек қана компьютер жағдайындағы символардың кейбір сандарының жиынтығы болып табылады, біз биттердің жиынтығын пайдаланатын боламыз. Сонымен, біз жібергіміз және қайта құрғымыз келетін хабарламалардың кейбір биталарын аламыз.

12345678901234567 # A1c8eF! 7ab8zzeiZ2 * Mr3 12345678901234567

Шығыс хабарлама

шифрленген хабарлама



Ашық кілт

жабық кілт



15.6 суреттегі шифрлеудің жабық кілті (жоғарғы) ашық кілтпен (төменгі) шифрлеуге байланысты.

N биттер, кілттің көмегімен. Біз мұны әр п-биталық блоктың жиынтығы үшін жасаймыз, жаңа хабарлама жинаймын. Қабылдаушы да сондай-ақ п-биталық блоктарда алған хабарламаны таратады, әр блоктың шифрі ашылған. Қарапайым мысал ретінде, хабарлама "Hello" деп қарастырайық. ASCII-де бұл хабарлама бес 8-биталық жиынтықтан тұратын бола-

ды (біз 0-ден 7-биталық ASCII символын әр символ 1 байт алатындай етіп қосамыз). Бұл бізге береді: 01001000 01100101 01101100 01101100 01101111. Енді біз бұл 40 битті аламыз да п-биталық жиынтыққа бөлеміз. Біздің мысалымыз үшін біз пайдаланатын $p=13$. Бұл бізге биталардың төрт жиынтығының орнына беріледі: 0100100001100 1010110110001 1011000110111 1. Соңғы жиынтық тек қана бір бит болғандықтан, 10000000000000 болатындай 0s подушка айналырамыз.

Кілт әрбір блокты қосарланған сан түрінде қарастырады. Ол оны басқа санға айналыру үшін қосарланған санның кейбір математикалық операцияларын пайдаланатын болады. Жол табудың біріне бүтін сандардың тізімін пайдалану жатады, мысалы, 1, 2, 5, 9, 19, 41, 88, 181, 370, 743, 1501, 3003, 6081. 13 бүтін сан бар болғандықтан, егер сәйкес бит 13-биталық жиынтықта 1 болатын болса, біз бұл бүтін сандардың әрқайсысынан бірге қосамыз. Мысалы, 0100100001100 үшін, біз 2, 19, 743, және 1501 қосамыз бірге (себебі тек қана 1 бит блокта екінші, бесінші, 10 және 11, және 2, 19, 743 және 1501 екінші, бесінші, оныншы, және 11 бірізділік болып табылады). Сонам, біздің кілтіміз, біз жіберуге арналған қосарланған файлды кері конверттейтін 0100100001100-ді 2265-ке айналырды.

Біздің 13 бүтін саннан тұратын бірізді жиынтығымыз өсу тәртібімен ғана емес, одан асып түсетін барлық сандардың сомасынан да көп екеніне назар аударыңыздар. Мысалы, $88 > 4119 + 9 + 5 + 2 + 1$. Бұл біздің номерімізді жеңіл шифрлеу үшін өте маңызды. Номердің шифрін ашу үшін кілттегі санға аз немесе тең санға анағұрлым жақын сандарды табу талап етіледі, бұл мағынаны есепке ала отырып, 0 жеткенше қайталаймыз. Осылайша, 2265 шифрін ашу үшін біз келесіні орындаймыз:

Көбірек белгі ≤ 2265 : $1501 \ 2265 - 1501 = 764$.

Көбірек белгі ≤ 764 : $743 \ 764 - 743 = 21$.

Көбірек белгі ≤ 21 : $19 \ 21 - 19 = 2$.

Көбірек белгі ≤ 2 : $2 \ 2 - 2 = 0$. Done.

Осылайша, біз 2265-ті 1501, 743, 19 және 2 шештік, немесе 0100100001100 қосарланған белгі. Ашық кілтпен шифрлеу үшін, біз кейбір қосымша математиканы пайдалануымыз керек. Біріншіден, біз номерлердің алғашқы тізімін (біздің құпия кілт) ашық кілтті жасау үшін пайдаланамыз. Ашық кілттердің саны мұндай қасиетке ие емес. Мысалы, ашық кілт 642, 1284, 899, 1156, 643, 901, 1032, 652, 1818, 940, 2266, 552, 723 болуы мүмкін. Біздің ашық кілтіміздегі номерлер, позициядағы біздің жеке номерлерге айналып, екі, кейбіреулері көптеген жай сандардан қосымша мағынаны орындайды.

Біз өзіміздің ашық кілтті әлдебіреуге бере аламыз. Ашық кілтті есепке ала отырып, дәл сол процесс берілген қосарланған блокты анықтау үшін қолданылады. Біздің алдыңғы бинарлы блогымыз үшін (0100100001100), сәйкес номерлерді ашық кілтте қосу керек, жабық кілтпен емес. Бұл сандар тағы да екінші, бесінші, 10 және 11-ші сандарға сәйкес, 1284, 643, 940 және 2266 болып табылады. Бұл сома 5133. 2265 жіберудің орнына біз 5133 жібереміз. Ашық кілттің көмегімен 5133 шифрін шешу өте қиын, себебі біздің көпшілік маңызды сандарымыз аддитивті қасиетке ие емес. Негізінде, сандар ретсіз болғандықтан, шифрді шешу үшін біздің ашық кілттегі белгілердің барлық мүмкін болатын комбинацияларын теруді талап етеді. Осы 13 санның 213 түрлі комбинациясы бар. Мысалы, біз көпшілік кілітінің алғашқы номерін өздігінен байқап көруімізге болады, ол 642. Бұл 5133 тең, сондықтан біз енді алғашқы екі санды байқап көре аламыз, $642 + 1284$. Бұл 5133 тең емес, осылайша біз келесі жолыбірінші және үшінші сандарды байқап көреміз, $642 + 899$. Бұл әлі 5133 емес, сондықтан біз алғашқы үш санды байқап көреміз, $642 + 1284 + 899$. Соңғы есепте, бізге дұрыс шешімді табу үшін барлық 213 түрлі комбинацияны байқап көруге тура келуі мүмкін.

Егер біздің шифрленген белгімізді ашу соншалықты қиын болса, қабылдаушы оны ашпақшы? Естеріңде болса, қабылдаушыда жабық кілт бар. Біз алынған алғашқы қатардағы сандарды екі жай сан арқылы қайта құруымыз керек. Бұл 5133-ті 2265-ке конверттейтін болады. Енді біз номерді жеңіл әрі тез ашу үшін өзіміздің жабық кілтті пайдаланамыз. 13 немесе 200 пайдаланудың маңызды түрлері блоктың өлшеміне тәуелсіз, барлығына шифрді ашу аз уақытты алады.

сын құру үшін, сондай-ақ кэштерді, фильтрлер мен кателерді түзету операцияларын (бақылау сомаларын) құру үшін ертеден-ақ пайдаланылып келеді. Идеяның маңызы бастапқы белгіні қабылдауда және оны бүтін санмен алмастыруда. Бұл ашық кілтпен шифрлеуді пайдалана отырып біз үшін жасалған, бірақ бізге адамның есімін айтуды қайта құру керек болса, ASCII символдарының белгілерін есімге қосар едік. Әрі қарай біз жай саннан бүтін санды айыру модулі бойынша операторды пайдаланамыз. Еске саламыз, модуль бойынша бөлу орындалады және бізге бүтін қалдықты береді. Егер біздің жай санымыз 1301 болса, онда $5133 \bmod 1301 = 1230$. Біздің жұмысымыздың нәтижесінде мод 0-ден шығатын жаңа сан және жай теріс сан болады (бұл жағдайда сан 0-ден 1300 дейін).

Бүгінгі күні пайдаланып жүрген шифрлаудың көп таралған стратегиялары бар. Бұл жерде солардың кейбіреулеріне тоқталамыз. Төменде жеке шифрлаудың барлық маңызды алгоритмдері берілген.

- DataEncryptionStandard (DES) - developed 1970 жылдары, бұл алгоритм блоктардың 56-биталық өлшемдерін пайдаланады. Оны қауіпсіздік бойынша Ұлттық агенттік пайдаланды, бірақ қазіргі уақытта қауіпсіз болып саналмайды.

- DES шығатын AdvancedEncryptionStandard (AES), ол тез шифрлау және шифрды ашуға арналған бағдарламалық және аппараттық құралдарда жеңіл жүзеге асырыла алатын, желілерді алмастыру-орын ауыстыруды пайдаланады. Ол блоктың 128-биталық өлшемін пайдаланады, бірақ кілттер өзіне 128 биттен 256 битке дейінгі 32-разрядты қадаммен өлшемді ала алады.

- TripleDES - DES бірнеше сағаттың ішінде бұзылуы мүмкін болғандықтан ол қауіпсіз деп саналмаса да, TripleDES тиісті хабарламаны шифрлау және шифрді қайта ашу үшін бірізді DES үш кілтін пайдаланады. Мысалы, егер K1, K2, K3 кілттер, және хабарламаны шифрлеу K3 қолдану жолымен жүзеге асырылса (K2-1 (K1 (хабарлама), бұл жерде K2-1 K2 хабарламаның шифрін ашу үшін пайдалануды білдіреді. Дешифрлеу сонан соң K3-1 көмегімен жүзеге асырылады (K2 (K1-1 (шифрленген хабарлама))). Әрбір клавиштің өлшемі 56 бит және 64-биталы блокта жұмыс істейді. Дегенмен, үш клавишті қолдану көрсеткендей, мұнда кодтың күрделілігі айтарлықтай өседі, оны бұзу анағұрлым күрделі. Шабуылға қарамастан, нәтиже заманауи компьютерлермен бұзылуы мүмкін 112-биталы кілттің эквивалентіне тең болады.

- Message-DigestAlgorithm (MDN) MD5 анағұрлым соңғы нұсқасы болып табылады. Бұл 128-бит шифрлеу алгоритмі, 32-белгілік он алтыншылық хэш-номер пайдаланатын хэш-функцияны пайдаланады. Дегенмен,

1996 жылдан бастап табылған шектеулі кілт өлшемі мен кемістіктердің арасында MD5 қайта қолданылмайды және SHA-1 алмастырылды.

- ША-0 және SHA-1 160-биталы блоктарға негізделген осындай алгоритмдер. MD5 сияқты, олар хэш-функцияларды пайдаланатын хабарлама алгоритмінің дайджесті. Оларда MD5 табылған кемшіліктер болмаса да, әдеттегідей өз әлсіз тұстары бар. Бүгінде SHA-2 федералды үкімет пайдаланады, өлшемдері данасына 224 биттен жоғары, төрт әртүрлі хэш-функцияларды біріктіру жолымен SHA-1 арттыру. SHA-3 2012 жылы енгізу сатысында болды.

Анағұрлым жиі қолданатын екі шифрлеудің маңызды алгоритмдері жариялы RSA және DSA болып табылады (сандық қолтаңбаның алгоритмі). RSA, 1978 жылы алгоритм жазған үш адамның құрметіне қойған, ашық кілттен жабық кілтке генерациялау үшін үлкен екі жай сандарды пайдаланады. Бұл екі жай сандар жабық кілтпен бірге сенімді қорғалуы керек. АДС RSA алгоритміне қарағанда анағұрлым күрделі болып табылады. АДС ашық кілтпен шифрлеу үшін АҚШ үкіметі пайдаланатын стандартты алгоритм болып табылады.

Бұл алгоритмдер түрлі бағдарламалар мен хаттамаларда қолданылады. Бұл жерде біз оларға қысқаша тоқталамыз.

- EPW (WiredEquivalentPrivacy) 1999 жылы, шифрленген сымсыз эквивалентті сымды желіде табуға мүмкіндік беретін құрал ретінде шығарылды. Жиырма алты 10-разрядты белгілерді кілт ретінде пайдалана отырып, көптеген кемшіліктері бар екені анықталды және бірінші WPA-ға, соңынан WPA2 алмастырылды.

- WPA және WPA2 (Wi-FiProtectedAccess) хаттамалар сымсыз желілердегі ресурстар арасындағы коммуникацияларды шифрлеу үшін пайдаланады. WPA2 WPA-ға шамамен 2004 алмастырылды. WPA2 AES алгоритміне негізделген. WPA және WPA2 хаттамаларды жіберуге дейін шығарылған сымсыз жабдықтармен жұмыс істеуге негізделген, сондықтан хаттамалардың анағұрлым ескі жабдықтармен кері байланысы бар. WPA және WPA2 сияқты қауіпсіздіктердің белгілі бір кемшіліктері бар, әсіресе Wi-FiProtectedSetup атауы бар функцияны пайдаланған кезде. Бұл функциядан бас тарта отырып, WPAжәне WPA2 анағұрлым қауіпсіз болып табылады.

- SSH Linux-та қауіпсіз қабаты болып табылады, Telnet, Rlogin және RSH байланыстарының қауіпсіз емес қабаттарын алмастырушы. Ол шифрлеуді ашық кілтпен пайдаланады. Бірнеше нұсқалар бар, ережеге сәйкес, SSH деп аталады (немесе SSH-1), SSH-2, және OpenSSH. Telnet портр 23 пайдаланғанда, SSH порт 22 пайдаланады. SSH сонымен қатар

SCP-дан файлдарды қауіпсіз жіберуді қамтамасыз ету үшін қолданылады (көшіріп басудан қорғалған). SSH альтернативті формасы SFTP болып табылады, файлдарды жіберудің қауіпсіз хаттамасы. Бұл FTP SSH сессияның ішінде пайдаланудан ерекшеленеді.

- SSL және TLS Secure Socket Layers болып табылады және сәйкесінше, Transport Layer Security. TLS айтарлықтай дәрежеде SSL алмастырады. Бұл байланыс хаттамалары TCP / IP-де жоқ шифрлау/дешифрлау мүмкіндіктерін бере отырып, TCP / IP тасымалдау деңгейін жақсарту үшін пайдаланылады. Байланыс шифрлеудің құпия кілтін немесе ашық кілтпен шифрлеу көмегімен орындалуы мүмкін. Соңғы жағдайда, кілттер / IP TCP қолдасуы кезеңінде және сервермен жіберілетін сертификат арқылы ұсынылады. SSL және TLS дауысты IP және VPN байланыс бойынша жіберуде пайдаланатын HTTPS, SMTP (электронды почта) қосқандағы түрлі хаттамалар қатарында қолданылады.

- HTTPS порт 80 орнына порт 431 арқылы HTTP қауіпсіз формасы болып табылады. HTTPS серверден клиентке сертификатты жіберуді талап етеді (15.4 суретке қайта оралу). Сертификат екі тәсілмен қолданылады. Біріншіден, оның клиент хабарламаны шифрлей алатындай ашық кілті бар. Екіншіден, егер оған сертификаттау органы қол қойған болса, ол оның түпнұсқасын тексереді, оған сервер мәжбүрлейді. Сертификат ескірген, өздігінен келісілген және қол қойылған сертификат, немесе сертификат барлығында бірдей сұрау салмайды веб-браузер клиентке ескерту жібермейді.

Заңдар

Біз бұл тарауды АҚШ үкіметі тарапынан әрекет ететін бірнеше маңызды заңнамалық актілерді қарастыра отырып қараймыз, ақпараттық қауіпсіздікке олардың ықпал етулерінен екенін атап өту керек. Дегенмен бұл толық тізім емес, сондай-ақ заңдардың анағұрлым толық тізімі бар веб-сайттар емес, тарауда келтірілгенін әрі қарай оқисыздар. Бұл тізімге өзге елдердің заңнамалары енбеген, бірақ сіздер Канададағы және Құрама Штаттардағы аналогтық заңдарды табасыздар.

- Құпиялылық туралы Заң 1974 – заңнамалық органның дербес мәліметтерді жинауға және пайдаланудағы ең бастапқы бөлімінде қатысты, бұл акт федералдық үкіметтің әрекетін шектейді. Заңда былай делінген: Ешбір ведомство жеке тұлғаның дыбыс жазушыға қатысты жазбаша сұрауына сәйкес немесе алдын ала жазбаша келісімін қоспағанда, кез келген тұлғаның немесе мекеменің байланыс құралының көмегімен

жазба жүйесіндегі кез келген жазбаны ашпауы тиіс *

Заң сонымен қатар, федералдық ведомстволар өздері жинайтын мәліметтерді, олардың себептерін жариялы ұсынулары тиіс, сондай-ақ олардың үстінен жинаған мәліметтерге назар аударту, олардың тұтастығын қамтамасыз ету үшін кез келген адамның мүмкіндігін бекітеді. Халық санағын жүргізу бюросы және құқық қорғау органдары сияқты заңнан алып тасталғандар бар.

- Отбасы туралы заң құқықтарға және құпиялылыққа үйретуге 1974 (FERPA) – студенттерге өздерінің жеке білім беру құжаттарына қол жеткізуге, бұл жазбаларға өзгерістер енгізуге құқық береді, дәлсіздік немесе мүлде жоқ екені туралы мәліметтер болуы керек, және алдын ала рұқсатсыз студенттердің жазбаларды таратудан мектептерді сақтау қажет. Бұл егер студенттерге бұл расымен де рұқсат етілмесе, оқушылардың ата-аналарымен (мысалы, сыныптардың) студент туралы ақпаратпен бөлісу қабілетіне кедергі келтіретіндіктен, маңызды акт. Бұдан бөлек ол аспирантураны немесе дербес жұмыс берушіні стенограммадан босату үшін студентті автор етуді талап етеді.

- Сыртқы бақылау (барлау) туралы Заң 1978 – бақылаулар Құрама Штаттарында жүргізілгенде электронды бақылауларды пайдалану үшін бекітілген стандарттар мен процедуралар.

- Құпиялылық туралы Заңның электронды коммуникациялары 1986 – жедел – іздестіретін сыртқы барлау туралы мұндай Заң, компьютерлік жүйелер арқылы (телефон арқылы әңгімелесулерді тындаудың басқа формаларының ішінде) электронды тындауға арналған ережелер мен талаптарды бекітеді.

- 1988 жылғы қорғау туралы Заңның құпиялылығы мен Matching Компьютер – мәліметтер базасын жеткізу және түрлі мәліметтер базасы арасындағы мәліметтердің сәйкестігі үшін басқа бағдарламаларды пайдалануды шектеу жолымен 1974 жылғы жеке өмірге қол сұқпау туралы Заңға өзгерістер енгізеді. Бұл актісіз, бір агенттік нақты бір жеке тұлғаға арналған профиль құру үшін басқа мекемелерден алған мәліметтерге дербес қол жеткізе алады.

- 1994 жылғы қорғау туралы Заңның құпиялылығын жүргізушілер, мемлекетке жүргізуші куәлігінен жүргізушілерді тіркеу барысында жиналған мәліметтерді (мысалы, мекен-жайлар, құнды құжаттардың әлеуметтік нөмірлері, бойы, салмағы, көзінің түсі, фотосуреті) сатуға тыйым салады.

- 1996 жылғы медициналық сақтандырудың жауапкершілігі туралы Заң. Басқа стратегиялар арасында денсаулық сақтаудың электронды жазбаларына арналған ұлттық стандарттың портативтілігі (HIPPA) және қа-

жеттілігі (мысалы, емдеуші дәрігердің қызметкерге қоятын сақтандырудың шарты бар шектеулер). FERPA, HIPPA сияқтылар агенттіктерге осы тұлғаның келісімінсіз медициналық-санитарлық көмекті таратуға тыйым салады. Бұл

* Сіз Әділет министрлігінің веб-сайты туралы толық заңды <http://www.justice.gov/opcl/privstat.htm> таба аласыз.

Неге бүгін дәрігерге бару керек себептерінің бірі, ең алдымен құпиялық мәселелерін шешетін құжаттарды толтыруды талап етеді.

- Digital Millennium Copyright 1998 жылғы Заң (DMCA) бүкіл әлемдегі компьютермен цифрланған материалды таратуға авторлық құқықты бұзатын, оны заңсыз ететін, авторлық құқық туралы екі шарт. Бұл заңға авторлық құқықпен қорғалған материалдарды веб-сайттарға орналастыруды немесе жүктеуді заңсыз ететін ережені енгізеді, алғашында материалдарда, мәтінге дыбыс, бейнежазба немесе бағдарламалық код болды ма белгісіз. Бұдан басқа, заң қылмыскер үшін Интернетті қолжетімді ететін интернет-провайдерлерден жауапкершілікті алып тастайды және осы акт бойынша қылмыстық іс қозғалды. Осы заңның нәтижесінде YouTube түрлі қолданушылардың YouTube сайтында авторлық құқықпен қорғалған акпараттарды орналастырғаны үшін жауап бермейді; Бірақ, нәтижесінде YouTube авторлық құқықтар иегерінің өтініші бойынша авторлық құқықпен қорғалған материалдарды өшіруге жиі тапсырыс береді.

- Сандық қолтаңба және 1998 жылғы электронды Authentication туралы Заң - (SEAL) –permits қаржы операцияларындағы куәландырылған сандық қолтаңбаларды пайдалану және қолтаңбаның сандық механизмдерін сақтауды талап етеді. Ол сонымен қатар сандық қолтаңбаларда қолданылатын криптографиялық алгоритмдерге стандарттарды қояды. 2 жыл өткен соң оның соңынан электронды қолтаңбаларды ұлттық саудада және Global қолдана бастады.

- 2002 жылдан бастап Заңның Қауіпсіздігі Бұзушылығы Ескертуі, америкалық штаттардың көпшілігі қандай да бір компаниядан (соның ішінде коммерциялық емес ұйымдар мен мемлекеттік мекемелер) кімнің жазбалары бұзылған, жоғалған немесе ұрланған болуы, соның ішінде мұндай мәліметтер шифрланбаған немесе компромисті түрде болса, бұл туралы барлық тараптарды хабардар етуді талап ететін заңдар қабылдады. Мысалы, егер файлдар тасымалданатын компьютерде сақталса, және бұл компьютер жоғалып кетсе дербес жазбаларды, файлдарда сақтаған тұлға әрекет туралы ескертілуі тиіс.

- 2004 барлауды реформалау және терроризмнің алдын алу туралы Заң – мұның қаншалықты мүмкін екенін талап етеді, арнайы қызметтер ақпараттармен алмасады, Пер – дербес террористік қауіптер бар ақпараттар жиналды және онымен алмасуда. Бұл акт 9/11 террористік шабуылдың алдын алу үшін барлау қауымдастығының сәтсіздігіне тікелей жауап болды.

Жоғарыда келтірілген актілер заңдарға қосымша АҚШ үкіметі криптографиялық технологияларды экспорттауға шектеу қойды. Мысалы, жеке тұлға немесе компания бұзылуы өте қиын белгілі бір криптографиялық алгоритмдерді сата немесе алмаса алмайды.

АҚШ Үкіметі. Үкімет сондай-ақ көптеген заңдарды жаңарту үшін, оларды компьютер дәуіріне енгізу үшін заңнамалық жинақ қабылдады. Мысалы, компьютердің ұрлануы ұрлыққа жатады, қандай да бір санатқа жатпайды. Сонымен қатар Интернетке және электронды почтаға қатысты цензураны қамтитын заңдар да бар.

16. Ақпараттық технология сала-сындағы мансап

ИТ-мансап

Оқулық өзі басталған жерінде аяқталады: ИТ-мансапты және ИТ-саламен байланысты мансапты зерделеу жолымен. Бұл тарауда ИТ түрлі салалары әрбір салаға талап етілетін нақты рольдерге басымдық бере отырып қарастырылады. Бөлімдердің арасында ИТ-адам ұмтылатын сертификаттар формалары және жақындағы жалақылары бірлескен болып табылады. ИТ студенттер ретінде мүмкін, өз мансабына көп ой жұмсамаған болар, олар колледжді бастаған кезде оқырманның қызығушылығын арттыра алатын мәтінді түрлі тақырыптармен сипаттайды. Оларға ИТ-этикалар, сондай-ақ сандық бөліну сияқты сандық әлемдегі әлеуметтік мәселелер, интернет-цензуралар, және интернет-тәуелділік сияқты тақырыптар жатады. Тарау ИТ-адам үшін үздіксіз білім берудің түрлі формаларын талқылаумен аяқталады.

Аталған тарауды оқытудың мақсаты болып табылады

- ИТ саласындағы мансапты талқылау.
- ИТ саласындағы үздіксіз білім беру формаларын сипаттау.
- ИТ-мен байланысты әлеуметтік мәселелерді енгізу.
- Ақпараттық технологиялар саласында өзін ұстаудың этикалық нормаларын талқылау.

Сіз ИТ ақшаға бола қызығасыз ба? Мүмкін. Қанша дегенмен, сіз технологияларды тұтынушы ретінде өстіңіз. Сіздің қызығушылығыңыз әлеуметтік бұқаралық ақпарат құралдарының әсерінен немесе жұмыстан немесе мектептегі компьютер ойындарынан басталуы мүмкін. Немесе сіз достарыңыз бен отбасыңыздың әсерінен қызығушылық таныттыңыз. Қызығушылығыңыз қандай болмасын, бұл сізді технологияға алып келді, сіз оған мүдделісіз, себебі сіз технологияларды жақсы көресіз. Сіз, бәлкім мәселелерді шешудің сараптамалық сауалнамасынан өткен боларсыз. Бұл ИТ-мансапқа апаратын жалпы жол. Сіздің бақытыңызға орай, ИТ саласындағы мансап жеңіл қолжетімді ғана емес, жалақысы

жоғары және қанағаттандыра алатын мансапқа жеткізе алады! Бұл жерде біз ИТ-мансаптарға бағыт алатын боламыз. Соңынан бұл тарауда біз сіздердің ойларыңызға келмеуі мүмкін Ит-мансаптың кейбір аспектілерін қарастырамыз. Технологиялардан өзге, өз білімін үнемі жаңартудан және өз қабілеттерін жақсартудан басқа сіздердің алда-рыңыздан шығатын мәселелер бар: құқықтық, этикалық, әлеуметтік. Ит мансабының жалақысы жоғары, ал келешектегі жұмыс орындарының өсуі ұзақ уақыт бойы өте көп болады. 16.1 кесте, жұмыс орындары саны мен айлық жалақының өсуі мансабы компьютермен байланысты түрлі салалар үшін ақпаратты көрсетеді. Компьютерлік бағдарламашының позициясы бағдарламалық қамтамасыз ету алаңының қысқаруына назар аударыңыздар және бағдарламалық қамтамасыз етудің сұранысынан бағдарламалық қамтамасыз етуді әзірлеушілерге сұраныс арта түседі.

Таблица 16.1 АҚШ Компьютер Мансап Жұмыс Өсім (сандар мыңмен)
басы средняя 2008

	2008	2018 (Est.)	Inc.	Inc. %	зарплата
бағдарламашылар	426.7	414.4	-12.3	-2.87	\$69,620
Компьютерлік бағдарламалық обеспечение инженеров, приложений	518.4	689.9	175.1	34.01	\$85,430
Бағдарламалық қамтамасыз ету жүйесінің инженерлерін компьютерлік бағдарламамен қамтамасыз ету Компьютерное программное обеспечение инженеров системы	394.8	515.0	120.2	30.44	\$92,430
Сараптаманың компьютерлік жүйесі	532.2	640.3	108.1	20.31	\$75,500
Мәліметтер базасының әкімшілері	120.4	144.7	24.4	20.26	\$69,740
Сеть жүйелері және компьютерлік жүйелер	339.5	418.4	78.9	23.23	\$66,310
	292.0	447.8	155.8	53.36	\$71,100

16.1 кестеде келтірілген мансаптардың ішінде соңғы үш мансапты ИТ қарастыра алады. Мәліметтер базасының әкімшілері, жүйелік әкімшілер мен сеть әкімшілері 2018 жылға дейін жоғары қарқынмен өсуі байқалады, және бұл қарқын болашақта бәсеңдейді деуге негіз жоқ. Бұл жерде келтірілмеген басқа мансаптарға веб-бағдарламалар, әкімшінің веб-сайттары, жүйелік архитектор және компьютер қауіпсіздігі бойын-

ша маман кіреді.

Бұл мансаптардың барлығы техникалық шеберліктің жоғары дәрежесін талап етеді. Бұл дағдыларға компьютерлік жүйелерді, операциялық жүйелерді, бағдарламалық қамтамасыз ету мен аппарат құралдарын түсіну кіреді. Бұдан басқа, ИТ-қызметкер мәселені шешеді деп күтілуде. Сонымен қатар олар өмір бойы дербес болулары қажет. Бұдан өзге, барлық ИТ-мамандар үшін жазбаша және ауызша коммуникацияларда, топтың бірлескен жұмысында және жетекшілік етуде тәжірибелі болуы, ИТ-этика жөнінде нақты білуі және өзінің дағдысы мен білімін үздіксіз жетілдіріп отыруға деген ниеті өте маңызды. Іскерлік тәжірибедегі іргетасы ИТ-маманның жұмысының алдағы мансабына көмегі тиюі мүмкін. Бүгінде ИТ-қызметкер жалдауға ниетті ұйымдардың көпшілігі ITEDy-университеттік бағдарламаларының 4 курс бітірушілерін құп көреді. Бұл бағдарламалардың бітірушілері техникалық шеберліктерге бағытталған оқыту катионына қарағанда (2 жыл техникалық, кәсіптік-техникалық немесе жергілікті колледжде) анағұрлым кең білім алғандар. Дегенмен, 4 жылдық IT оқыту сирек кездеседі, сондықтан ұйымдар ИТ-қызметкердің жұмысына қарап отырып, кейде компьютерлік ғылымдар, іскерлік әкімшілік немесе ақпараттық жүйелерді басқару (ИС) және информатика сияқты аралас пәндерге де қарайтын болады. Студенттер, ИТ-профильге дәрежесі сәйкес келмейді, әйтсе де қалаулы бола алады, егер олар компьютерлік жүйелерде, сетьтерде, мәліметтер базасында және басқаларда тиісінше курстық жұмыстар алса немесе сәйкес тәжірибе жинаса.

Бұл мансаптардағы жауапкершілік деңгейі ұйымның өлшеміне байланысты. Үлкен ұйымдарда сіз жауапкершілігі тек қана ИТ-мамандық саласында ғана болатын тұлғаларды кездестіргіңіз мүмкін, мысалы, жүйелік қауіпсіздіктің немесе веб-сервердің әкімшісі болу. Шағын ұйымдарда көбінесе бірнеше ИТ-рольдерді бір адам атқаруы мүмкін. Бұл жерде біз ИТ-маманы алуы мүмкін позициялардың типінің және осы позициялардың әрқайсысының міндеттерінің әрқайсысына тоқталамыз. Сонымен қатар біз сертификаттау формаларын (егер мұндай болса) және айлық жалақыларын қарастырамыз.

Жүйелік әкімшілік жасау

Жүйелік әкімшілер компьютерлік жүйе құрайтын жабдықтарды орнатуға және техникалық қызмет көрсетуге жауап береді. Жүйелік жабдықтарға коммутаторлар, маршрутизаторлар, концентраторлар, сетьаралық экрандар, қауіпсіздікті бейімдеу құрылғылары, сымсыз қол жеткізу нүктелері мен IP (VoIP) телефондар бойынша дауыс жеткізу сияқты

мамандандырылған жүйелік құрылғылар кіреді. Жүйелік бағдарламалық қамтамасыз етуге осы аппарат құрылғыларына орнатылған түрлі операциялық жүйелер кіреді, протоколдар, сондай-ақ қосымшалар сеть арқылы хабарланады. Бұл өзіне маршруттау протоколдарын, қол жеткізу тізімдерін, порттарды агрерлеу протоколдарын, ағаш протоколдарын қамтитын, виртуалды локальды жүйелерді, виртуалды жеке жүйелерді, IPv4, IPv6 мен шабуылдауды анықтау жүйелерін қосады. Жүйелік әкімшілер мұндай бағдарламалық қамтамасыз етуді орнатуға, жөндеуге және ұқыпты ұстауға жауапты болуы мүмкін. Жүйелік әкімшілер сондай-ақ жүйені қорғауды талап ете алады; мұндай міндет жүйелік әкімшілер сияқты басқа да ИТ-мамандармен үйлесе отырып орындалуы мүмкін.

Егер сеть баяуласса немесе сөніп қалса, жүйе әкімшісі өзін жоғары қысымды іздеушінің және жоюшының роліне қоюы мүмкін. Әкімші мәселені диагностикалау, оң шешім табу және оң шешімнің дұрыстығын анықтау үшін өзінің білімі мен тәжірибесін пайдалануы тиіс. Шешім аталған мәселені пайдаланушының көзқарасы тарапынан тиімді шешуге, тиімді қолжетімді болуға және бағасы қымбат болмау қажет. Бұдан өзге, мәселелер мен оның шешімдері жақсы құжаттандырылуы тиіс.

Деңгей жүйесін әкімшілеудің жоғары позициясын кейде жүйелік немесе жүйелік инженерлердің архитекторлары деп атайды, ұйымның қажеттілік көлемі мен барынша күшіне жауап береді. Бұл жұмысқа сәйкес жүйелік жабдықтарды таңдауды және жүйелік мекендерді барша ұйым масштабында таратуды жоспарлау кіреді. Жүйелік әкімшілер TCP / IP ағымының барлық базалық білімін, сондай-ақ Interconnect (OSI) ашық жүйелерінің моделін ұстанатын басқа да жүйелік протоколдарды талап етеді. Сонымен бірге жүйелік әкімші жүйелік қауіпсіздік, сымсыз басқару, VoIP, интернет-провайдерлерді басқару сияқты мамандықтарға, сондай-ақ жүйенің сенімділігі мен күшін жоспарлауға жиі назар аударады. Осы міндеттер тізімінен көрініп тұрғандай, жүйелік әкімшілер өз ролінде жетістікке жету үшін даярлық пен тәжірибенің орасан көп мөлшерін талап етеді. Технологиялардың сипатының өзгеріп тұруынан, жүйелік әкімшілер сауда журналдары, конференциялар, жарияланымдар мен үздіксіз білім беру арқылы жаңа технологияларды үнемі зерделеп отыруы тиіс. Түрлі мансаптық сертификаттаулар әкімшілік жүйені жаңартуға құнды толықтыру болып табылады. Анағұрлым талапқа сай сертификаттаулардың арасында CCNA (Cisco Certified Network Associate), CCNP (Cisco Certified Network Professional), Novell CNE (Certified Novell инженер), CWNA (Certified Wireless Network Administrator) және CompTIA Сеть + болып табылады. Жүйелік әкімшілердің айлық жа-

лақылары (2008 жылғы бойынша) \$ 41000 дейін және \$ 104000 дейін жоғары орташа айлық жалақы \$ 66310. Қазіргі уақытта компанияның жүйелерін басқару кілтіне, кез-келген деңгейдегі мектептерде жұмыс істейтіндерге, аналогтық жұмыс орындарына қарағанда анағұрлым жақсы төлейтін телекоммуникациялық операторлар үшін жұмыс істегенше \$ 3000 артық төлейді.

Жүйелерге әкімшілік ету

Сеть әкімшілері сетьтегі аппараттық құралдарға және бағдарламалық қамтамасыз етуге жауап беретін болса, жүйелік әкімшілер осы сетьтегі ресурстарды орнатуға және оны ұстап тұруға жауап береді. Жүйелік әкімшінің ролі аппараттық құралдарды, бағдарламалық қамтамасыз ету мен жүйенің конфигурациясын қамтиды. Компьютерлерге мэйнфемдер, ERSминикомпьютерлер, сондай-ақ дербес компьютерлер, жүйелік серверлер кіруі мүмкін. Жүйе әкімшілері сонымен қатар бірнеше операциялық жүйелермен таныс болуы керек, себебі ұйымда бірнеше платформадағы компьютерлер болуы мүмкін.

Міндеттер алуан түрлі және ұйымның тәуелділігіне байланысты нұсқаларға бөлінеді. Жүйелік әкімші кем дегенде ұйымның автоматтарында операциялық жүйелерді орнатады және жөндейді, жаңа жабдықтарды орнатады, бағдарламалық қамтамасыз етуді және техникалық қызмет көрсетуді модернизациялауды жүзеге асырады, пайдаланушыларды әкімшілеуді өндейді, резервтік көшіреді және қалпына келтіреді, сондай-ақ мониторинг жүйесін орындайды, жөндейді және ақауларды жояды. Жүйе әкімшісі сонымен қатар қосымшаларды әзірлеумен және қауіпсіздік жүйесі мен сенімділікті кеңейтумен байланысты болуы мүмкін, және сеть жүйесіндегі файлдық жүйемен жұмыс істеулері керек. Жүйе әкімшісіне кейбір бағдарламаларды оболочка сценарийлерінің көмегімен орындауға тура келеді, сондықтан қайталанатын міндеттерді автоматтандыруға болады. Бағдарламалардың көптеген мөлшері өлшеміне және ұйым қызметкерлеріне байланысты қажет болуы мүмкін. Саясатты әзірлеу және ИТ-инфрақұрылымды жоспарлау жүйелік әкімшішлерге салынған міндеттер болуы мүмкін.

Жүйе әкімшісінің ролі ұйымның өлшеміне және қажеттілігіне байланысты жүйе әкімшісінің талаптарын жабуы немесе толық шығаруы мүмкін. Шағын ұйымдар ИТ-қызметкерлерді аз жалдайды, соның әсерінен міндеттері алуан түрлі болуы мүмкін. Басқа жағынан қарағанда, шағын ұйымдар шағын сетьтердің әсерінен қажеттіліктері, компьютер ресурстары аз, сондықтан жүйелік әкімшінің назарын аударатын жағдайлар аз. Білікті жүйе әкімшісі кез-келген қолжетімді ве-форумға назарт ауда-

рады, жүйе әкімшілері ретінде ағымдағы мәселелерді талқылайды және шешу жолдарын іздейді.

Сетьтегі әкімші сияқты жүйе әкімшісі де компьютерлік жүйелер тиімді және сенімді жұмыс істеуіне кепілдік беру үшін үлкен даярлық пен тәжірибені талап етеді. Жүйе әкімшісі оның әрі қарай дамуы үшін белсенді роль атқаруы қажет.

Жүйе әкімшісі ретінде сертификаттау үшін бірнеше мүмкіндіктер, сондай-ақ сауда журналдары, конференциялар мен жарияланымдар бар. Сертификаттау мүмкіндігіне MCSA Microsoft (Microsoft Certified Systems Administrator), MCSE (Microsoft Certified Solutions Expert) және MCITP (Microsoft Certified IT Professional), Red Hat, RHCE (Red Hat Certified инженер) және RHCSS (Red Hat Certified Security маман) және CNA компаниялары Novell (Novell Certified Administrator), сонымен қатар бұған дейін еске салған жүйе сертификаттары кіреді.

Жүйе әкімшілері үшін айлық жалақы негізгі жүйе әкімшісінің деңгейінде қалады. Осы жұмыс орнының екі санаты қажеттілікті арттыру санының артатынын көрсетеді, кем дегенде 2018 жылға дейін ондаған мың жаңа позициялар ашылатын болады.

Веб-әкімшілік

Веб-әкімші, сондай-ақ веб-мастер деп белгілі, бірақ нақтырақ атағанда веб-сервердің әкімшісі веб-сайттардың жағдайына жауап береді. Бұл веб-сайттарды дамытудан ерекшеленеді (басқа бір-бірімен байланысты мамандықтардағы WebDeveloper қара). Негізінен веб-әкімшіге веб-сайттарды орнату, баптау, қолдау, қорғау және кемшіліктерін жою қажет. Веб-сервер бағдарламалық қамтамасыз ету болып табылады, ол бір немесе бірнеше компьютерлерде жұмыс істейді. Веб-әкімші аппараттық құралдар үшін жауапты болады немесе болмайды. Әдетте, веб-әкімші қандай да компьютердің жүйелік әкімшісі болмайды. Осылайша, веб-әкімші жүйелік әкімшімен орнату мен конфигурацияның бірнеше аспектілері бойынша жұмыс істеуге мәжбүр болады.

Конфигурация веб-парақшалар орналасатын, директиваларды көрсету үшін конфигурациялық файлдарды құруды өзіне қосады, қолжетімділікті авторландырудың қандай да талаптары бар ма, қауіпсіздіктің қандай нысандары жүзеге асырылады, және қандай құжаттар кодталады, мысал үшін, шифрлауды қолдану арқылы, сонымен қатар сервер орындай алатын тілдер сценарийі.

Веб-сервердің өнімділігінің мониторингі әкімшіге сервердің параметрлерін нақты баптауға мүмкіндік береді. Бұл мысалы, файлдарды қысуды қосып көбірек қуаттылықты немесе бірнеше қатты дисктерді орнатуды,

және сценарийлер сервері жағында сандарды азайтуды өзіне қосады. Сонымен қатар мониторинг автоматты өндірілетін лог-файлдар көмегімен әкімшіге URL тапсырысын береді, олар клиентті ізденіс тәртібінен алынған мәліметтер болуы мүмкін.

Веб-сервердің қауіпсіздігін қамтамасыз ету бизнес үшін қиын болып келеді. Веб-әкімші кілт сөздер файылдарын құрастыру және серверлік скрипттер сақталуы және тексерілуі мүмкін болатын орындарды анықтау үшін қажет болуы мүмкін. Қауіпсіздік веб-сервердің бағдарламалық қамтамасыз етілуінің көмегімен тікелей жүзеге асырылуы немесе операциялық жүйе, немесе қандай да бір комбинациялар арқылы қосылуы мүмкін. Қауіпсіздік сұрақтарын мәліметтердің интеллектуалды талдауы үшін пайдаланылатын журналдар файлдарын талдау арқылы анықтауға болады.

Веб-сервердің бағдарламалық қамтамасыз етілуі әртүрлі шағын бағдарламалардан құралады. Бұл бағдарламалар URL қайтабағыттаудағы қателерді режимдік өңдеуді тіркеу үшін URL сұраныс файлдарының ізденісімен байланысты, серверлік тапсырыстардың кез-келген санын өңдейді. Apache сияқты ақысыз веб-серверлер бар, және де IBMWebSphere, OracleWebLogic және MicrosoftIIS сияқты көпшілікке мәлім коммерциялық веб-серверлер бар. Веб-әкімші бағдарламалық қамтамасыз етілуді түсіну, таңдау және сәйкесінше аппараттық жүйелерде бағдарламалық қамтамасыз етуді қалай қолдау туралы нұсқауларды әзірлеу үшін, веб-сервердің бағдарламалық қамтамасыз етілуін таңдауға қатыса алады немесе қатыса алмайды. Аппараттық шешімдер көптеген компьютерлік серверлермен физикалық OUS арасындағы бөлу серверін, бірнеше мекен-жайлар серверін өзіне қосады, және сақтау жүйесінің қандайда бір типін пайдалану қажет. Веб-сервердің өнімділігінің мониторингі кезінде веб-әкімші аппараттық құралдарды қайтақұру және жетілдіру бойынша негізделген ұсыныстар қабылдай алады.

Веб-сервердің әкімшілері өзі үшін немесе құрастырушыларды қолдау үшін қосымшаларды әзірлеу үшін немесе орындалуын талап ете алады. Қолдау модульдердің қосымша веб-серверлерін орнату түрінде болуы мүмкін, Perl немесе PHP кодтарын орындай алатын модульдер сияқты. Онда сонымен қатар даму үрдісіне әсер ететін, саяси шешімдер болуы мүмкін, мысалы басқа компьютерде веб-серверден тыс орындалатын барлық сценарийлер немесе барлық осы скриптер CGI-BIN каталогында сақталады.

веб-сервердің әкімшілігі Интернетте болуды қалайтын ірі ұйым үшін жұмыс істеуі мүмкін, сонымен қатар қабылдаушы веб-сайттар компаниялары үшін жұмыс істеу мүмкіндіктері бар. осындай жағдайда, бір сер-

вер әртүрлі компаниялардың бірнеше веб-сайттарын орналастыру үшін пайдалануы мүмкін. Әкімші веб-сервердің өнімділігін қамтамасыз ету және веб-парақшаларды әзірлеушілердің бірнеше топтарымен жұмыс істеу бойынша қосымша мәселелерге ие болады.

Веб-мастерлердің еңбек ақысы қатты ауытқуы мүмкін \$ 40000-дан \$ 90000 дейін, орташа есеппен алғанда шамамен \$ 55000. Әкімшілер үшін қолжетімділігі төмен веб-сертификаттаулар бар. Біріншілік нысан Certified Administrator Internet Webmaster Server ретінде белгілі. Бұл көптеген CIW аббревиатуралылардың бірі, бірақ CIWS үлкен бөлігі веб-әзірлеумен байланысты.

мәліметтер базаларына әкімшілік ету

Веб-әкімшілер мен жүйелік әкімшілер тәрізді мәліметтер базасы әкімшілері маңызды міндеттердің ұзын тізіміне ие, бірақ мәліметтер базаларын басқару жүйелерін (МББЖ) жобалау, әзірлеу және қолдау акцентімен. Басқа әкімшілік аудандармен сияқты міндеттер орнатуды, қызмет көрсетуді, өнімділіктің талдауын және қателерді жоюды өзіне жатқызады. Жүйелік әкімшілер мен желі әкімшілерінен бөлектенген, МББЖ байланысты есептік жазбалар мен үрдістерді құрастыру, өнімділік баптаулары параметрлері, арнайы қосымшалар бар. Мәліметтер базасы әкімшісі сондай-ақ ескі жүйелерден жаңасына мәліметтерді интеграциялаумен байланысты болуы мүмкін. Мәліметтер базаларының көптегені немесе басым бөлігі енді Интернет арқылы қолжетімді болғандықтан (немесе, тым болмағанда, локалды есептегіш желілерінде), әкімші сонымен қатар қашықтықтан қатынасу үшін тиісті қауіпсіздік шараларын қамтамасыз етулері қажет.

МББЖ көптеген алуантүрлі типтері бар. Аса көпке белгілілері реляциондық мәліметтер базалары болып табылады, мысалы Oracle, MySQL немесе Microsoft SQL Server. Бірақ NoSQL мәліметтер базасы және бұлтты есептеулерге бағдарланған мәліметтер базаларын есепке алып, мәліметтер базалары технологияларының басқа типтері өсіп келе жатыр. Техниканың барлық облыстарында сияқты, мәліметтер базаларының жаңа технологияларының өсуінің жалғасуы, үнемі оқу және өзін-өзі жетілдіру сызбасында болатын мәліметтер базалары әкімшілерін талап етеді.

Веб-әзірлемелерге қатыспауы мүмкін веб-әкімшілерге қарағанда, мәліметтер базаларының әкімшісі қосымшаларды әзірлеуде және дамытуда белгілі бір орын алатын болады. Ол сол қосымшаларды тиімді қолдау үшін қажетті мәліметтер базаларын жобалау қосымшаларын әзірлеушілерге көмек көрсету арқылы болуы мүмкін. Ол үшін дағдылардың ерекше жиынтығы қажет, олар мәліметтер базасын үлгілеуді,

сызбаны, нормалауды әзірлеуді және өнімділікті баптауды өзіне жатқызады. Мәліметтер базалары әкімшілерінің аса мамандандырылған сипатына байланысты басқа ИТ-рөлдердің еңбек ақысы жоғарыда аталғандардыкінен жоғары. Орташа орта еңбек ақы \$ 70 000 құрайды, ал еңбек ақының ең жоғарғы деңгейі \$ 110 000 артық болады. Сертификаттар алдымен МББЖ нақты бағдарламалық қамтамасыз ету болып табылады, мәліметтер базасы әкімшісі Microsoft Certified сияқты, Oracle Certified, және MySQL сертификаттары.

Компьютерлік қолдау маманы

Компьютерлік қолдау маманы қандай да жалпылаушы термин болып табылады. Әрине, қолдау мамандарының кейбір рөлдері жүйелік әкімшінің жауапкершілігі болып табылады. Қолдау қызметінің маманы ең алдымен ақауларды жою бойынша тренер ретінде жұмыс істейді. Оқу міндеттері материалдар өндірісінен құралуы мүмкін, мысалы техникалық нұсқаулықтар, құжаттамалар және оқу сценарийлері. Оқыту көптеген нысандарда болуы мүмкін, топпен оқыту, жекелеп оқыту, оқыту видеофильмдері, сонымен қатар оқыту видеофильмдерін және бағдарламалық қамтамасыз етуді өндіру.

Өзіне ақпараттық қызмет жұмысын (төменнен қара), әскери қызметкерлерді ақпараттық қызметке оқыту, және персоналдың ақпараттық қызметін бақылауды қосады. Ол сонымен қатар өзіне жүйелік әкімшісі (лері) және желі әкімшісі (лері) сияқты ИТ басқа персоналына көмекті де қосуы мүмкін. Компьютерлік қолдау маманы үшін міндеттер соншалықты үйреншікті болуы мүмкін, жүйеге қалай кіру немесе қандай да бір міндет арқылы пайдаланушыны аттап өтіп бағдарламаны қалай қосу, желілік файлдық серверге немесе принтерге қалай қосылу, немесе жүйені күрделі ашу, неге ұйымдық мүшелер ресурсқа қолжетімділікке ие болмайды деген сияқты.

Компьютерлік қолдау маманы бастапқы деңгейде қызмет атқарады. Ол телефон қоңырауларына және пайдаланушылардың сұрағына электронды хатпен жауап береді. Қойылған сұрақтардың типтері адам жұмыс істейтін ұйымға байланысты болады. Мысалы, егер сіз Интернет қызметін жеткізуші қызметінде жұмыс істесеңіз, сіз Интернетке қосыла алмайтын немесе қосылу әлсіз болған клиенттерден телефон қоңырауларын қабылдайсыз. Ақпараттық қызмет ретінде сіз жалпы сұрақтарға жауап беретіндей жағдайда болуыңыз қажет, және сирек кездесетін сұрақтарға жауапты қалай таба алатын немесе аса мамандандырылған білімі бар адамдарға клиенттерді бағыттай алу жағдайында болу қажет. Сіз сонымен қатар сізді күлдіретін сұрақтарды естуіңіз мүмкін. Кейбір білімсіз (технологиялық айтқанда) адамдармен сөйлесуге әзір болыңыз!

Міне ақпараттық қызметтің шың әңгімелесулерінің кейбір мысалдар (шамамен).

- «Сізде қандай компьютер?»

- «Ақ»

-

Клиент: «Сәлем, бұл Роза. Мен дискетамды шығара алмай отырмын»

Қолдау қызметі: «Сіз батырманы басып көрдіңіз бе?»

Клиент: «Ия, әрине. Ол қысылып қалды»

Қолдау қызметі: «Бұл жаман. Мен жазып алайын»

Клиент: «Жоқ. Бір минут күте тұрыңызшы. Мен оны әлі салған жоқпын. Ол әлі менің үстелімнің үстінде екен. Кешіріңіз»

Қолдау қызметі: «Экранның сол жағындағы «Менің компьютерім» батырмасын басыңыз»

Клиент: «сол жақ немесе менен сол жақтағы ма?»

Қолдау қызметі: «Қайырлы күн! Сізге қалай көмектесе аламын?»

Клиент: «Сәлеметсіз бе, мен басып шығара алмай отырмын»

Қолдау қызметі: «Сіз «бастау» батырмасын баса аласыз ба?»

Клиент: «Тыңда достым... маған дауыс көтермеңізші! Мен Билл Гейтс емеспін, сіз білесіз ғой!»

Клиент: «Сәлем, қайырлы күн, бұл Марта, мен баспадан шығара алмадым. Мен тырысқан сайын, ол: «Принтер табылмады» деп жазады. Мен тіпті принтерді көтеріп, монитордың алдына қойдым, бірақ компьютер әлі де оны таба алмағаны туралы айтады»

<http://www.funny2.com/computer.htm~~pobj> сайтынан алынған.

Қолдау қызметі: «Сіздің кілттік сөзіңіз кішкентай «а» әріпінен басталады, алма сөзінде сияқты, бас әріп «V», Викторда сияқты, және «7» саны.

Клиент: ««7» бас әріп болып табыла ма?»

Клиент: «Менде өте үлкен мәселе бар. Жолдасым, менің компьютерімнің бастапқы бетіне сурет қойды, бірақ мен тышқанды қозғалтқан сайын, ол жоғалып кетеді!»

Қолдау қызметі: «Сізге қалай көмектесе аламын?»

Клиент: «Мен өзімнің алғашқы электрондық поштамды жазып отырмын»

Қолдау қызметі: «Жақсы, сонда қандай мәселе?»

Клиент: «Менін адресімде «а» әріпі бар, бірақ оның айналасына шеңберді қалай қоямын?»

Жұмыс үстеліне қашықтықтан қатынасудың көмегімен, компьютерлік қолдау маманы пайдаланушы компьютерін қашықтық режимінде «бақылауға ала алады». Бұл маманға мәселелерді жақсы түсінуге, анықтауға және жоюға мүмкіндік береді. Ол сондай-ақ қиыншылық сезінетін адам атынан шамалы еніп маманға жұмыс істеуге мүмкіндік береді. Бұл егер адам компьютерді немесе туындаған мәселені білмегенде, уақытты үнемдеуге мүмкіндік береді.

Әкімшілдік карьераға қарағанда, қолдау мамандары 4 жылдық деңгейге, қолдау қызметінде жұмыс істеу үшін қажетті дағдылардан басқа арнайы дағдыларға ие бола алмайды. Осыған қарамастан, 4 жыл деңгейі барлар, акция иеленеді де аса қиын және, мүмкін, аса жағымды міндеттерге ауысады. Еңбек ақыларда айырмашылық бар, қолдау қызметінің мамандары толық жұмыс күнінде және толық емес жұмыс күнінде жұмыс істей алады. Окладтар диапазондары \$ 25000 және \$ 70750 қарағанда жоғары бола алады, бірақ қолдау мамандары көбінесе \$ 30000-нан \$ 45000 дейін диапазон жасайды.

ИТ басқару

мамандандырылған техникалық дағдыларға күшті және жиі қойылатын қосымша талаптар, осында аталған барлық ИТ-карьералар үшін, әңгімелесуді және бірігіп жұмыс істеуді қажет етеді. Бұл жұмсақ дағдылар ИТ-мамандарына соңғы пайдаланушылармен, қосымшалардың әзірлеушілерімен, сонымен қатар жоғарыдағы Басқарушымен жақсы жұмыс істеу мүмкіндігін береді. Бұл қазіргі кезде жұмыс берушілердің не себептен тек техникалық дайындықпен қатар, жан-жақты білім алуды қамтамасыз ететін, 4-жылдық ИТ-деңгейі бар, потенциалды қызметкерлер арасынан ИТ-персоналды іздеу себептерінің бірі.

Тұлғааралық қатынасты, лидерлікті, және бизнес-тәжірибені түсінуді көрсете алатын бұл ИТ-мамандары ИТ басқарудың болашақ позициялары үшін жақсы мүмкіндік бола алады. ИТ-менеджер бұл жоюға қажетті мәселелерді түсіну үшін ИТ-инфрақұрылымның техникалық жағын, сонымен қатар техникалық персоналды бақылауды жүзеге асыру мақсатында бизнесті басқару жағын түсінетін және жоғарыдағы бастықтармен еркін араласатын адам. ИТ-менеджер жоба менеджері болады, ол шекті кезеңдерді бекітуі, сыртқы құрастырушылармен жұмыс істеуге, мүмкін бюджетті өндеуге және ұсынуға, сондай-ақ басқаруындағы жобаға (ларға) қатысты шешімдер қабылдауға тиіс. Сонымен қатар жұмыс

скерлерді жалдау және жұмыстан босату да ИТ-менеджердің міндеттерінің бөлігі болуы мүмкін.

Басқару позициялары, әдетте, осыған дейін сипатталған техникалық облыстардағы еңбек ақыға қарағанда аса жылдам жылдамдықпен. Егер соңғы есепте сіздің мақсатыңыз басқарудың осындай позициясын иелену болса, сіз колледжде оқу уақытында мүмкіндіктеріңізді арттыра едіңіз. Бұл мүмкіндіктер студентті басқаруға қатысатын колледждің клубтары мен ұйымдарына қосылудан, сонымен қатар жобаларды басқару, экономика, бухгалтерлік есеп, және басқару сияқты бизнес-облыстарда қосымша сабақтарды игеруден құралады. Қызметі бойынша әрі қарай жылжуды ізденетін студент, бірнеше рет ілгері бизнес-деңгей, немесе CIT магистр дәрежесі сияқты ілгері техникалық дәреже үшін MBA немесе MIS сияқты мектептерге оралады. ИТ-менеджерлер техникалық қызметкерлерге қарағанда еңбек ақыларын потенциалды екі есе ете алады.

4 жылдық ИТ дәрежесін мазмұнымен жабатын басқа да карьералар бар. ИНГ тізімінде олардың кейбіреуі талқыланады.

Компьютерлік қылмыстар

Әдетте, компьютерлік соттық сараптама карьерасына қосылатын адам, немесе компьютерлік криминалистика дәрежесіне ие болады, немесе қылмыстық істердегі қосымша фоны бар жоғары техникалық білімге ие болады. Техникалық білімі информатика, ақпараттық технология немесе компьютерлік инженерия болуы мүмкін. соттық-медициналық сараптаманың компьютері жаңа және дамушы сала болып табылады, және осы тақырыпта толық оқу жоспарын ұсынатын бакалавриаттың бірнеше бағдарламалары бар. Осыған қарамастан, көптеген компьютерлік ғылымдар және ИТ-бағдарламалар компьютерлік криминалистиканы қосады. Сонымен қатар, компьютерлік криминалистика немесе жақын қауіпсіздік бағдарламасының түлектері өсу деңгейінде.

ИТ немесе ЭЕМ ғылыми бағдарламалары саласында алынған техникалық дағдылардан басқа, операциялық жүйелер, компьютерлік қауіпсіздік, компьютерлік жады, бағдарламалық қосымшалар, криптология салаларында тәжірибелік білімдер, және мүмкін, бағдарламалау мен инженерлік шешімдерді бағдарламалық қамтамасыз етудің екі тілін білу қажет. Сонымен қатар күшті аналитикалық дағдылар маңызды болып табылады.

Компьютерлік криминалистикада жұмыс істейтін адам тек компьютерлік жүйемен қамтамасыз етілген немесе бұзу және санкцияланбаған қолжетімділік дәлелдерін іздеумен ғана айналысатын болып табыл-

майды. Оның орнына компьютерлік қылмыстар орын алған кездерде дәлелдерді құқыққорғау органдарына ұсыну қажет, олар сізді жұмысқа жалдауы мүмкін. Осылайша, сіз істі қалай құрастыруды және соттық тәртіпте дәлелдерді қалай анықтауын, жинақтауын, сақтау және ұсынуын түсінуіңіз қажет.

бүгінгі таңда компьютерлік сот сараптамасында жұмыспен қамтылу мүмкіндігі жоғары шығар. Жұмыс істеудің келесідей нұсқалары бар, тікелей құқыққорғау органдары қызметкерлері үшін жұмыстар, құқыққорғау органдарымен кеңестер, немесе банк және корпорациялар сияқты өз мәліметтерін қорғауда және құқық бұзушыларды қудалауға мүдделі ұйымдар үшін жұмыс істеу. Еңбек ақы, қазіргі таңда, ИТ басқа мамандықтарына қарағанда елеулі жоғары, жылдық окладтар \$ 75,000- \$ 115,000 диапазонында, ал орташа \$ 85000 болады. Осыған қарамастан, компьютерлік сот-медициналық сараптамаларда белгілі бір дәрежесіз сіздің ұсыныстарыңыз, 4 жылдық толық дәреже алған студенттерге қарағанда шамамен шектелген болуы мүмкін.

Веб-әзірлеме

Веб-әзірлемеді ойнауға болатын көптеген рөлдер бар. Ұйым өзінің компаниясын «Интернетте жоғары» көтеру үшін бір адамды немесе компания веб-парақшалар әзірлеушілерінің штатын жалдауы мүмкін. Сондықтан веб-әзірлеуші атқаратын ерекше орыны компания мен жағдайларға тәуелді ауытқып отыруы мүмкін. Бірақ веб-әкімшілікте талқыланған веб-сайттың әкімшісіне қарағанда, веб-әзірлеуші әртүрлі дағдыларға ие болады.

веб-әзірлеушілердің дағдылары HTML, CSS (стильдердің каскадтық кестелері) және сценарийлердің әртүрлі тілдерін пайдалану мүмкіндігінен тұрады. Сценарилер JavaScript, Ruby, Python, Java-апплеттер, Active Server Pages (ASP), Visual Studio .NET (Visual Basic, C ++ немесе C # қоса алып), Perl және PHP әзірленуі мүмкін. Осыған қарамастан, веб-әзірлеушінің осы барлық тілдердің білуін күтпейді, веб-әзірлеуші олардың кейбіреуін ғана пайдалануы да мүмкін. Веб-әзірлеушінің бағдарлама-лау аспектісі маңызды болғанымен, веб-әзірлеуші компьютерлік ғылым дәрежесін міндетті талап етпейді (дегенмен көптеген компьютерлік ғалымдарды веб-әзірлеуші ретінде көруге бейімбіз).

Веб-әзірлеушіге дұрыс түсті, орналасуды, графиканы, анимацияны және т.б. пайдаланып эстетикалық веб-парақшаны құрастыру бойынша білім қажет болуы мүмкін. Веб-әзірлеуші сонымен қатар Интернетпен байланысты көптеген техникалық және әлеуметтік сұрақтарды түсінуі қажет. Мысалы, байланыс желісі, өткізу қабілеттілігін, дискті көшіру жылдамдығы табиғатын түсіну, сонымен қатар прокси-серверлерді

пайдалану веб-әзірлеушіге аса тиімді және оңай жүктелетін веб-сайт құрастыруға көмектеседі. Әлеуметтік мәселелерді түсіну веб-сайттың шабуылдардан қауіпсіздігін қамтамасыз етуде шешуші мәнге ие болуы мүмкін, веб-сайттың тұтастығын қамтамасыз ете отырып, сонымен қатар веб-сайттың мазмұны саяси әдептілікті болуын қамтамасыз етеді. Сонымен қатар қолжетімділіктің бірнеше мәселелері бар, сондықтан алуан түрлі кемшіліктері бар адамдарға кіріп және еш қиындақсыз орнында маневрлеуге болады. Қазір федералды заңға сәйкес, үкіметпен қаржыландырылатын федералдық агенттіктер мен ведомстволардың сайттары гандикапты адамдар үшін қолжетімді болуы тиіс.

Веб-әзірлеменің өзі бірнеше міндеттерден құралады, олар графикалық дизайн және көркемөнер, бизнес, мәліметтер базаларына қолжетімділік, телекоммуникациялар, адамның компьютермен қарым-қатынасы және бағдарламалау. Негізінен, веб-қосымшаларды әзірлеу сервер жағында скрипттерді жазу және динамикалық веб-парақшаларды қолдау үшін басқа серверлік қосымшалар және сервермен және мәліметтер базасымен өзара байланыстан құралады. Бүгінгі күндері веб-әзірлеушінің позициялары үлкен сұранымға ие. Басқа жағынан, оларда еңбек ақының ірі диапазоны бар, мүмкін, өз компаниялары үшін веб-портал құрастырғысы келетін компаниялардың кең спектрін шағылыстыратын шығар. Еңбек ақы \$ 30 000 сияқты соншалықты төмен болуы мүмкін, және орташаға қарағанда жоғары \$ 70000 диапазонында болуы мүмкін. Сонымен қатар өз статусын жақсарту үшін қажетті көптеген қолжетімді сертификаттар бар. Оларға Microsoft Certified Professional Developer, CIW Web Developer бірнеше сертификаттары ("Web Administration" қаpa), HTML Developer сертификациясы және Sun Certified Web Component Developer жатқызылады.

Бағдарламалау/инженер-бағдарламашы

компьютерлік бағдарламашылар кез-келген пәнде санынан, ғылымды, математиканы және бизнесті қоса, келгені тарихи қалыптасқан. Информатика дәрежесі 1980 жылдан бастап танымал бола бастады. Бүгінгі таңда, компаниялар бағдарламашыларды жалдамайды, бағдарламалардың компьютерлік ғылымдары (немесе, кей жағдайларда компьютерлік техника) түлектерін жалдайды. Бағдарламалық инженерияға акценттің артуымен кейбір мектептерде инженерлік компьютерлік бағдарламалар бар, олар көбінесе бағдарламалардың компьютерлік ғылымына ұқсас. Компьютерлік ғылым пәні ИТ қарағанда фундаменттің басқа типін қамтамасыз етеді. Компьютерлік ғалым үшін математикалық негіздеуде есептелімділік және шешілімділік тақырыптарын оқу қажет. Бұл

тақырыптар ғалымдарға өлшеу құралдарын береді, берілген мәселенің шешімі халькопирита лэнга ретінде (немесе ол жалпы шешімі бар болып табыла ма). Ғалым сонымен қатар компьютерлік жүйенің негізгі табиғатын түсіну үшін компьютердің және операциялық жүйенің ұйымдастырылуын зерттейді. Ғалым сонымен қатар алгоритмдердің кең спектрін және мәліметтер құрылымын зерттейді және бағдарламалау тілдері тізімінде бағдарламаларды өткізеді.

ИТ-тұлға жағдайында сияқты, инженер-бағдарламашы / бағдарламалық қамтамасыз етуші білім шегімен қатар жүруі тиіс. Соған қатамастан, ИТ-маманы елеулі дәрежеде жаңа операциялық жүйелерді, олардың ерекшеліктерін, қауіпсіздік құралдары мен шараларын игерген кезде компьютерлік ілім бағдарламалаудың жаңа тілдері мен әдістерін игеретін болады. Бағдарламашылар үшін сертификаттар бар, бірақ 4-жылдық деңгейлі білім шегі барларға сертификаттардың қажеттілігі жоқ. Еңбек ақылар 16.1 кестеде көрсетілгендей, әдетте, ИТ-менеджментті есепке алмағанда, ИТ басқа салаларына қарағанда компьютерлік ғылым / бағдарламалық инженерия саласында жоғары.

Курстық жұмыстың аса қатты ісер етуші қоспаларының бірі, ИТ-дәрежелерді ғылыми компьютерлік курстық жұмыспен біріктіру болып табылады. Бұл бағдарламалаудың тамаша фундаменті бар ИТ адамды қамтамасыз етеді. Басқа жағынан, операциялық жүйелер, желілер және компьютерлік қауіпсіздікті нақтырақ түсінуі үшін қосымша ИТ-курстарды қабылдауда ғалым жақсы қызмет алады.

Ақпараттық жүйелер

15 бөлімде келтірілгендей, IS компьютерлерге қарағанда ұзағырақ болды. Мәні бойынша, ақпараттық жүйе дегеніміз, ұйымды басқаруды қолдау үшін пайдаланылатын мәліметтер мен ақпаратты жинақтау болып табылады. Көбінесе, ұйым бизнес болып табылады, біл осылай болмауы керек болғанмен. IS шегі белгілі бір деңгейде компьютерлік ғылыммен байланысты, IS компьютерлік құрылғыны, бағдарламалық қамтамасыз етуді және бағдарламалауды игеруден құралады. Осыған қатамастан, бағдарламалық қамтамасыз етуге акцент бизнес үшін бағдарламалық қамтамасыз ету айналасында айналады (мысалы, МББЖ, электрондық кестелер, статистикалық бағдарламалық қамтамасыз ету, транзакцияларды өңдеу үшін бағдарламалық қамтамасыз ету), сонымен қатар аппараттық құралдарды игеру көбінесе желілер айналасында, және азырақ сақтауды айналады. Пән көптеген бизнес тақырыптарды қамтиды, бухгалтерлік есеп пен экономикаға кіріспені қоса алғанда, және көбінесе басқару саласындағы бірнеше курстардан құралады.

Жұмыскер зерттеу жүйелерін әзірлеу үшін жобалық басқару мақсатында

ұйымдағы рөлдердің кез-келген санын өзіне ала алады. Жоба менеджері ретінде, адам жобаны жоспарлау, жобалау, орындалуы, мониторингі мен бақылауына жауап береді. Жоспарлау және жобалау кезеңдерінде менеджер спецификацияларды, сонымен қатар уақыт және бағалау құралдарына шығындарды өндіруі керек. Менеджер жобаны қолдау үшін қызметкерлерді жалдау үшін қажет болуы мүмкін. Фазаларды орындау, мониторингі және бақылауы кезінде менеджер үрдісті бақылау, мәселелерді анықтау және қосымша ресурстар алу арқылы мәселелерді шешу үшін жобаны жүзеге асырушы топтардың әрқайсысымен координацияланады. Диспетчер қызметкерлердің техникалық дағдыларына ие болуы тиіс емес, бірақ менеджердің негізгі технологияларды түсінуі үшін аса маңызды.

дамытуда еңгізу командасының бөлшегі болып табылатын IS қызметкері бар. Жүзеге асыру жоғары деңгейлі бағдарламалау тілін (мысалы, Visual Basic, COBOL немесе Java) қажет етуі мүмкін, бірақ компьютерлік кодтың нақты бөлшектерін үлкен деңгейде абстракциялау арқылы тұлғаны қамтамасыз ететін құралдарды пайдаланатыны аса ықтимал. Бұл құралдардың көбісі тілдердің 4-ші ұрпағы (4GLs) ретінде қолданылады, бұл бағдарламалау тілдері есептегіш техниканың төртінші ұрпағы кезінде әзірленгенін мензеп. Бұл тілдер есеп берулер генераторын іске қосады, мәліметтер базаларын басқару жүйелері мен генераторларын құрастыру, мысалы Excel, Access, SAS, SPSS, RPG, Mathematica және құралдар, ал жақын уақыттан бері XBase ++, XQuery және Oracle Reports. IS үлгілеу әдістері мен IS құралдары қарастырылады. Үлгілеу кезінде, қандай да бір ұйымдастырушылық тәртіпті көрсету тәсілін анықтауға тырысады, IS ұйымдастыруда пайдаланылатын операция болсын немесе тәсіл, мұнда білім ұйымдастырушылық бірліктер арасында немесе олардың арасындағы қандай да бір үрдістермен таратылады. Үлгіні есепке ала отырып, зерттеуші әдістерді анықтайды. Үлгі және әдістер жеткілікті нақты болуы қажет, сондықтан зерттеуші жауалап алу үлгісі мен әдістері күтімді тәртіп екенін көрсету үшін, ұйым үрдістерін жеткілікті деңгейде үлгілеуге мүмкіндігі бар. Үлгіні ескере отырып, зерттеуші үрдістерді жетілдіру және ұйым саясаты бойынша шараларды анықтауға талпына алады.

Компьютерлік маман

Компьютерлік техник компьютерлік құрылғыны тікелей орнату, жөндеу және ақауларды жоюға байланысты жұмыс істейді.

1. Сен компьютерді басқа адамдарға зиян келтіру үшін пайдаланбауын керек.
2. Сен басқа адамдар компьютерлерінің жұмысына кедергі жасамауын керек.
3. Сен басқа адамдардың компьютерлік файлдары айналасында тыншылық әрекет етпеуін керек.
4. Сен компьютерді ұрлық жасау мақсатында пайдаланбауын керек.
5. Сен компьютерді жалған куәлік беру үшін пайдаланбауын керек.
6. Сен ақшасы төленбеген бағдарламалық қамтамасыз етуді көшіру немесе пайдаланбауын керек.
7. Сен басқа адамның компьютерлік ресурстарын рұқсатсыз немесе тиісті өтемақысыз пайдаланбауын керек.
8. Сен басқа адамдардың интеллектуалдық шығуын
9. Сен жазып жатқан бағдарламаның немесе әзірленуші жүйенің әлеуметтік салдары туралы ойлайтын боласын.
10. Сен әрқашан компьютерді жақындарына көңіл бөлу мен сыйластықты сақтандыру арқылы пайдаланасын.

16.1 сурет Компьютерлік этиканың 10 өсиеті

* Компьютерлік әдеп институтында <http://computerethicsinstitute.org/home.html> веб-сайты бар, ол киберәдепті, 16.1 суретте аталған 10 өсиетті және басқа да көптеген мәселелерді талқылайды.

Инженерлер (IEEE). Бұл әдеп кодекстері бірінші кезекте бағдарламалаушылармен байланысты әрі кәсіби құзыреттілікті, өз жұмысының кәсіби шолуын қабылдай, келісімшарттарды ескерумен компьютерлік жүйе тәуекелдерін мұқият бағалауды қамтамасыз ете, өзінің есептеуіш техниканы және оның салдарларын түсінуін жақсартуға ұмтыла отырып, заңдарды түсінуді атап көрсетеді. Бұл кодтар әдептің ұғымдарын, олардың технологияға қалай қарайтынын, сондай-ақ кәсіпқой болу әдебін байланыстырады. Кез-келген ұйымның тиісті компьютерлік әдепке қатысты, әрі, дәлірек айтқанда, әкімшілердің рөліне қатысты шешу тиіс болған бірқатар мәселелері бар. Оларға жеке өмірдің қол сұғылмаушылығы, меншік құқығы, бақылау, дәлділік және қауіпсіздік жатады.

- Файлдарды қорғау - әкімші басқа адамдардың файлдарын қорғауды қандай жағдайларда қарастырғаны жөн? Қандай жағдайларда компания өз клиенттері туралы жинақталған ақпаратты пайдалануы тиіс?

- Иелену – кім жинақталатын деректерді және ұйым өндірген өнімді иеленеді?

- Бақылау – қызметкердің іс-әрекетін қандай дәрежеге дейін бақылау қажет?

- Дәлділік нақты бір қызметкерлердің немесе бүкіл қызметкерлердің құлау жауапкершілігі, дәлділігі кімге? Компания жұмысы деректердегі

қауіпті қателерді қаншалық жояды?

- қауіпсіздік, қандай да бір дәрежедегі, компания өз деректері мен жүйелерінің бүтіндігін қамтамасыз етеді?

Бұл бөлімде біз бірінші кезекте АТ-кәсіпқой үшін пайда болуы мүмкін әдептік дилеммаларға тоқталамыз. Содан кейін АТ-кәсіпқойлар ұйымда ұсына алатын саясаттың кейбір мәселелерін қысқаша қарастырамыз.

Келесі жағдайлар үшін, өзіңізді сол жағдайға қойып көріңіз. Не істер едіңіз?

- Бастығыңыз әлде біреудің компания ресурстарын порнографияны көру не жүктеп алу үшін пайдаланатынын көру үшін веб-серфинг қызметкерлерін бақылауда ұстауды сұрады. Сіз үшін қызметкерлермен ЧАТТА ЖАЗЫСУ әдепке жата ма?

- Бастығыңыз Сізден заңсыз төмендететін нәрселерді жүктегенін анықтау үшін нақты бір қызметкердің файл кеңістігін зерделеп шығуды сұрады. Бұл әдептілікке жата ма?

- Сіз кейбір уақытта жұмыста босай қаласыз. Осы кезде біліміңізді жаңарту үшін технологияларды пайдаланумен байланысты түрлі сайттарға кірдіңіз. Солардың арасында сіз компьютерлік жүйелерді бұзу жайында оқыдыңыз. Сіз үшін бұндай нәрселерді оқу әдепке жата ма? Сіз үшін жұмыс уақытында бұзу жайлы оқу әдептілік пе?

- Электрондық поштаның сервері тегеріш кеңістігінен орындалатынын байқадыңыз. Сіз компанияда жеке қажеттілігі үшін электрондық поштаны пайдаланатын қызметкерді іздеуді өз мойныңызға аласыз. Бұл әдептілікке жата ма?

- ұйымыңыз үшін прокси-сервер орнатылған. Бірінші кезекте сервер мақсаты уақытты үнемдей отырып, басқа пайдаланушылардың көргісі келіп кетуі мүмкін веб-беттерді кәштеу болып табылады. Дегенмен, прокси-сервер барлық өтініштер журналын да жасайды. Сіз үшін әріптестеріңіздің қандай сайттары және беттері қаралғанын көру үшін осы рұқсаттарды қарау әдептілікке жата ма?

- Сіз веб-әкімшісіз (веб-жасаушы емес). Компанияңыздың иелік ететін сайттарының бірінде сіздің жеке адам ретінде тіл тигізуші екеніңіз жазылған. Сізге не істеу керек? Егер ол қоғамдастық стандарттары бойынша ерсі болса, бұл әдебіңізді өзгертуге себеп бола ма? Егер бұл ақпараттың болуы қоғамдастығыңыз тарапынан ерсіз деп бағаланатынын сезінсеңіз, қайтесіз?

- Сіз веб-жасаушысыз. Сізден компания кебір веб-беттерді өздері сізге ұсынатын контентті пайдалана отырып, жүргізуді сұрады. Сіз бұл контенттердің мазмұны авторлық құқықпен қорғалған екеніне сенімдісіз. Не істеуіңіз керек?

- Сіз жеткізу үшін өнім дайындайтын ұйымда жұмыс істейсіз. АТ адамы бола отырып, енгізу командасының бір бөлігі болып табылмасаңыз да, Сіз өнімнің әлі дайын емес екенін білесіз, бірақ басшылық оларды өнімді қандай жағдай болмасын шығаруға мәжбүрлеп жатыр. Не істеуге тиіссіз?

- Бастығыңыз Сізден операциялық жүйені қауіпсіздік тұрғысынан осал жаңа версияға дейін жанартуды сұрап жатыр. Бұлай істеу әдептілікке жата ма?

- Сіз бастығыңыздың сізді кейбір нәрселер үшін жұмыстан шығарып бере ме деп қорқасыз (сезініп жүрген нәрселеріңіздің бәрінен ақталғаныз). Сақтық шарасы ретінде құпия шот (бэкдор) жасау мәселесін өзіңіз шешесіз. Бұл әдептілік пе? Заңды ма бұл?

- Ұйымыңыз адамдарға сайтта өздерінің Facebook беттерін жаңартуға еркін рұқсат етеді. Сіз қызметкерлеріңіздің кейбірі өзі туралы тым көп ақпарат салады деп уайымдайсыз. Осыған араласу сіз үшін орынды ма?

- Ұйымыңыз адамдарға жұмыста Facebook пайдалануға тыйым салатын саясатты ұстанады. Қызметкеріңіздің біреуі жұмыс уақытында Facebook бетін өзгерткенін байқап қалдыңыз. Бұл жөнінде хабарлайсыз ба?

Жоғарыда мысал ретінде келтірілген жағдайлардан Сізге жеке өмірдің қол сұғылмаушылығын немесе қызметкерлердің сенімін бұзатын жағдайларға өзіңізді қойып көру ұсынылған болуы мүмкін. Сізден қызметкерлерді бақылап жүруді сұрағанда, олардың артынан аңдуды сұрайды. Әдепті болу үшін көптеген факторларды ескеру қажет. Бұл факторларға компанияның қызметкерлерінің компьютерлері мен керек-жарақтарын пайдалану жөніндегі саясаты, сондай-ақ АТ-мамандардың рөлі кіреді. Жарияланған саясатсыз қызметкерлерді аңдуды ақтап алу қиын. Басқа жағынан қарасақ, тіпті компанияның саясаты болса да, бұл саясат қызметкерлерге қаншалық жақсы таныстырылған? Егер олар саясаттан бейхабар болса, оның пайдасы аз.

АТ-маман ретінде сіздің рөліңіз АТ-саясатын әзірлеп, енгізу болуы мүмкін (басқа жағынан саясат тек басшылық тарапынан ұсынылады). Полистер компанияның әдеп ұстанымын, сондай-ақ заңды бейнелеуі тиіс. Саясат АТ пайдаланудың барлық аспектілерін қамтуы қажет. Атап айтқанда, саясат мыналарды айқындауы тиіс:

- компанияда ойнайтын рөлін
- қызметкердің АТ дұрыс пайдалануын
- жұмыскердің жеке өміріне берілетін құқықтары (егер ондайлар болса)
- деректерді дұрыс пайдалануды
- деректердің қауіпсіздігі мен құпиялығын

- компанияда АТ көмегімен әзірленген идеяларға меншік құқығын (зияткерлік меншік)
- компанияның авторлық құқық мәселелерін қалай шешетінін

Саясатты әзірлеу АТ-маманы үшін бөтен болуы ықтимал. Бұл басқарумен айналысуға қарағанда жақсырақ болуы мүмкін. Дегенмен, АТ-маманы саясатты бұзбайтын шешімдерді қабылдау үшін ұйым саясатын жақсы білуі тиіс. Саясатты білмей, сіздің алдыңғы сұрақтарға берген жауабыңыз сізді тәуекелге жолықтыруы мүмкін.

АТ-маманының ұйым саясатын, сондай-ақ тиісті әдеп ұстанымын қалай түсінуі керектігі сияқты заңдарды да түсіну маңызды. Бастығыңыз сізден CD-R тегерішін табыстай отырып, бағдарламалық қамтаманы орнатуды сұрайтын жағдайды қарастырып көрейік. Бұл бағдарламалық қамтама заңға сатып алынған ба? Онымен бірге лицензиялары бар ма? Егер жоқ болса, сіз оны орнатасыз ба? Жұмыс берушіңіз тапсырманы орындауыңызды өтініп тұрғандығынан, бұл тапсырма әдепке сай немесе заңды дегенді білдірмейді. Заңды білсеңіз, өзіңізді қорғап қана қоймай, басшыңызға қауіпті заңсыз жағдайлар туралы хабарлай аласыз, Өкінішке орай, ақпараттық технологияларды және бағдарламалық қамтаманы реттейтін заңдарды түсіну қиын міндет болуы мүмкін. Тіпті заңдарды анықтай алсаңыз да, олар көбінесе түсініксіз әрі заң тілімен жазылады. Мысал ретінде мына жағдайды алып қарайық, компанияңыз саясат жасады, ол бойынша барлық қызметкерлердің әлеуметтік сақтандыру нөмірлері белгілі болады. Яғни, әлеуметтік сақтандыру нөмірі дерекхананы жазу үшін бірегей сәйкестендіргіш ретінде қызмет ететін болады. Осылайша, ол процес барысында олардың журналында да пайдаланылмақ. Өйткені бұл журналдағы олардың бөлігі болып табылады, әлеуметтік сақтандыру нөмірлері операциялық жүйедегі әлемдік есептелетін файлдарда сақталады (мәселен, / және т.б. / құпиясөз). Жүйелік әкімші ретінде сіз не істейсіз? Әлеуметтік сақтандыру нөмірін пайдалану компания саясатын бұзу болып табылады ма? Жоқ. Бұл заңды ма? Қатты айтпай. Бұл әдепсіз бе? Мүмкін.

Сіз басшылықты белгілі бір мақсатпен журнал үшін әлеуметтік сақтандыру нөмірін пайдаланудың жаман идея екеніне және сәйкесінше, бұл қорғалған деректердің кез-келген әлемдік есептелетін файлдардан өшірілетініне көндіріңіз. Дегенмен, бұл мәндер компанияның дерекханаларында қолжетімді, соның ішінде сіз бен әкімшілердің дерекханалары және дерекхана әкімшілері бұрынғыдай кіре аласыз. Бұл әдепсіздік пе? Жоқ, өйткені рұқсаты барлардың бұндай деректерге заңды қажеттілігі бар. Десе де, компания өз қызметкерлерін ақпараттың қауіпсіз

екеніне қалай сендіреді. Қорғалған деректерге рұқсаты бар адамдар осындай деректердің сезімталдығын түсінуге оқытылуы тиіс. АТ-мамандары портативті компьютерлер сынды осы құпия деректерді сақтайтын құрылғыларды сақтауға қосымша жауапкершілік арқалайды.

Flame және Stuxnet вирусы

Осы кітап қазіргі уақытта жарияланымға әзірленіп жатқандықтан, жаңалықтарда Flame деп аталатын жаңа вирустың шыққаны жайында айтып жатыр. Біріншіден, кейбір фондар. 2009 жылы Stuxnet вирусы (АҚШ аумағының, серіктестігінің израйылдық өнімі болып есептеледі) түрлі иран компьютерлеріне шабуыл жасаған, әсіресе Ирандағы уранды байыту бағдарламасына қатысатындарға. Stuxnet және соған ұқсас Duqu вирусы шамамен 500 КБ тартатын кішкентай вирус болған. Stuxnet пен Duqu вирустарының екеуі де саботаж негізінде жаңашыл зиянды бағдарлама ретінде жасалған.

Flame вирусы сондай мақсатты көздесе де, Ирандағы, сондай-ақ басқа да Таяу Шығыс пен Шығыс Еуропа елдеріндегі компьютерлерге жүргізілген талдау вирустың мүлдем басқа ұйым тарапынан жазылғанын көрсетіп отыр.

От жарыққа 2012 жылы ғана шықты, бірақ вирусты зерттеу кейбіреулерді кейбір компьютерлер 2007 жылдың желтоқсанында зақымданған деген ойға түйістірді.

От вирусы 20 МБ шамасында әрі Stuxnet вирусына қарағанда әлдеқайда күрделі. Оның негізгі мақсаты пайдаланушыларды аңду сияқты. Ол білдірмей деректерді (құжаттар, файлдар) ұрлауға, компьютер микрофоны қосылысымен айтылатын әңгімелерді жазып алуға, пернелерді басумен жазылатын сөздерді жазып алуға, Bluetooth арқылы көрші құрылғылардан телефон нөмірлерін алуға, скриншоттар жасап, сақтауға, сондай-ақ желі бойынша сәйкестендіру үшін табысталатын болғандықтан, пайдаланушылардың аттары мен құпиясөздерін көшіріп алу үшін локалды интернет-трафикті сканерлеуге қабілетті. Егер бұл жеткіліксіз болса, вирус адамдар үшін бэкдор есептік жазбасын жасауы да мүмкін, сондықтан олар көбірек вирус құрамбөліктерін жүктей алады. Вирус Интернеттегі 80 түрлі серверге қарағанда, жинақталған ақпаратты жалғыз табыстау үшін бағдарламаланған.

Flame вирусының өзі бірліктер жинағын білдіреді; бастапқы блоктар, компьютер зақымданған кезде, басқа бірліктерді жүктеу үшін қолданылады. Негізгі блокты жүктеуге, алуға, қаптауға және басқа құрамбөліктердің шифрын шешуге болады. Сонымен қатар, вирус, табылудан жасырынуға қабілетті сияқты.

От вирусы желі арқылы және USB-жинақтағыштар арқылы тарала ала-

ды. Отқа түрлі ұйымдардан бірнеше атау берілген, соның ішінде Wiper, Viper және Flamer.

Басқа әлеуметтік мәселелер

АТ-маманы АТ-шешімдерінің салдарларын түсінуі тиіс. Осылайша, ұйым саясатын қоспағанда, әдеп пен құқықтар үшін АТ жөніндегі маман шешуші әлеуметтік проблемаларды да түсінуі қажет. Біздің қоғам адамдары осы бөлімде сипатталған деген кейбір қауіптенулер.

Сандық алшақтық. Компьютерлер мен Интернетке қолжеткізуде кейбір олқылықтар бар. Бұл олқылықтар нәсілдің, жыныстың, табыс деңгейінің және географиялық орналасу орнының әсері. Гендерлік алшақтық көптеген елдерде едәуір қысқарды, бұл әсіресе әлеуметтік желілердің әйгіленуі есебінен. Бірақ технологиялық тұрғыдан дамыған әрі бай елдер мен кедей елдер арасындағы кемшіліктер өсуін жалғастыруда. Интернет ақпараттың ауқымды көлеміне қолжеткізуге мүмкіндігі барларға артықшылық беріп қойғандықтан, бұндай жағдайы жоқтар үшін ақпараттық өмір, саламатты өмір салты, білім мен жұмыспен қамту қол жетімсіз. Үкіметтер және халықаралық корпорациялар сандық технологияларға жүгінеді, бірақ ол бұрынғыша проблема болып табылады. Дегенмен, АТ-маманында қалай болса, сіз сандық алшақтық салдарларымен ешқашан да істі болмауыңыз мүмкін, осыны түсіне отырып, сіз ұйымыңызға дұрыс саясат жасауға көмектесе аласыз.

Денсаулықпен байланысты компьютерлік проблемалар. Компьютерді қарқынды түрде пайдалану денсаулықпен байланысты бірқатар проблемалар тудыруы мүмкін. Негізгі проблемалар қайталама күйзеліс жарақаттары (RSI) және көзге күш түсу. RSI ең көп таралған түрлерінің бірі білезікті туннельдік синдром, білезікке әсер ететін бұлшықет проблемасы. Бұл көбінесе пернетақта мен тінтуірді дұрыс пайдаланбаудың нәтижесінде туындайды. RSI-дан басқа, арқа бөліктері, иық мен шынтақтың ауыруы, тіпті санның және тізебуын деформациясын көбінесе компьютер салмағының ауырлығымен байланыстырады. Проблемалар тізбесіне сондай-ақ бас ауыруы мен күйзеліс кіреді. Бұл проблемалар жақсылап құжаттандырылған әрі оларды шешудің бірқатар шешімі бар. Эргономика, адам тәні және HCI (адамның компьютермен әрекеттесуі) жақсы таритын жабдық пен құрылғыны жобалауды зерттеу бізге жақсы жиһазды, көзге жеңіл компьютерлік мониторларды және компьютермен әрекеттесудің азырақ деструктивті түрлерін (мәселен, тінтуір орнына нүкте трек) берді. Эргономикаға қарамастан, адамдар баяғыша нашар келбеті себепті немесе олар компьютерді пайдалану кезінде үзіліс жасауды ұмытатындықтан, денсаулығына байланысты проблемаларға жо-

лығып жатыр. АТ мамандары эргономикалық шешімдерді түсініп, оларды есептеуіш ресурстарды сұрату кезінде пайдалануы тиіс.

Құпиялықты сақтау. Жеке адамның өміріне төнетін қауіп өсіп жатыр. Сіз туралы ашылып қойған көптеген жеке ақпараттар бар. Бұл сіздің туылған күніңіз, тұрғылықты жеріңіз, мінетін көлігіңіз, қашан және кімге үйленгеніңіз туралы фактілерді өз ішіне алады. Бұл ақпараттың көп бөлігі көпшілікке қолжетімді болады, өйткені ол өздігінен сіздің жеке өміріңіз үшін қауіп төндірмейді. Дегенмен, әлеуметтік желілердің, веб-сайттар мен блогтардың көмегімен адамдар жүйелі түрде көбірек ақпарат таратады.

Twitter пайдаланатын адамды қарастырып көрейік, ол отбасымен бір аптаға қаладан тысқары кетеді. Ақылды ұры үйді тауып алып, оның бір апта бойы бос болатынын білгендіктен, оны тонап кетуі мүмкін. Жеке дербес ақпараттың бұндай ашықтығы сандық төсеніш деп аталады. Осындай өзін-өзі жарнамалаудың қауіпін түсіну жеке деректерді ұрлаудың осы тұрғысынан барлық компьютер пайдаланушылары үшін маңызды болып табылады. Қауіп қызметкерлерін хабардар ету АТ-қызметкерлерге жүктелуі мүмкін. Дегенмен, бұл қауіптер тіптен маңыздырақ болуы да ықтимал. Адамдардан тікелей қорғалған ақпаратты алу үшін алаяқтар байқап көру фишингін пайдаланады. Бұған мысал телефон қоңырауы болады (немесе электрондық пошта бойынша). Бұл кезде авторландырылған АТ-адамының сізге хабарлау барысында ол тиісті деректерді (мәселен, әлеуметтік сақтандыру құпиясөзі немесе нөмірі) жоғалтып алған және сіздің абонентке осындай ақпаратты беруіңіз керек, әйтпесе аккаунтыңыз өшіріледі деп сендіреді.

Интернет тәуелділік. Бүгінде адамдардың компьютермен өткізетін уақыты олардың теледидар алдында өткізетін уақытынан асып кетті. Бұл біздің қоғамдағы керемет өзгеріс. Бұл қозғалысқа себеп болған нәрселердің бірі адамдардың электрондық қарым-қатынас түріне тәуелділігі болып табылады. Бұл электрондық байланыс түрлері электрондық пошта, әлеуметтік желі блоктары мен сайттарын өз ішіне алады. Осыдан келіп адамдарда интернетке тәуелділік ауруы пайда болды, тіпті кейбір адамдар үйінен тысқары жерде болғанда қолдарын смартфоннан алшақ ұстай алмайтын жағдайға жетті. Бұл көріністі біз адамдардың қоғамдық іс-шараларда (мәселен, кино, кешкі ас) болғанында байқаймыз, олар уақытының көбін адамдармен тілдесуге емес, телефондарын шұқылауға жұмсайды. Бұл нақты әлеуметтік проблема сіздің ұйымға тікелей әсер ете алмауы мүмкін. Дегенмен, көптеген жұмыс берушілер қызметкерлеріне әсер ететін диверсияның қамын жеумен жүр. Интернет-тәуел-

ділік еңбек өндірімділігіне елеулі әсер етуі мүмкін.

Ерсілік және цензура. Интернет адамдарға жер шарындағы барлық нәрсені дерлік оқып, карауға мүмкіндік беріп қойды. Өкінішке орай, Интернетке салынған нәрсенің бәрі бірдей адамдардың моральдық бағыттарына сәйкес келе бермейді. Интернетте парнографиялық сайттар, зорлық сипатындағы бейнебаяндар (бұндай бейнебаяндар YouTube сынды сайттарда бар) және наразы адамдарды әшкерелеу, осыларды адамдар ерсілік деп есептеуі мүмкін. Ерсілік бұрыннан бері қоғамымыздағы проблема болып келеді. Қызметтер рейтингі, Байланыс жөніндегі федералдық комиссия (FCC) және күзетші ұйымдар кино, теледидар мен радиоларды қадағалап отыруға тырысып бағуда. Алайда, Интернет кеңістігін оның жаһандық сипаты мен қорының өте көп болуы себепті қадағалау мүлдем басқа әңгіме. Цензура белгілі бір контентті немесе нақты дереккөздерден алынған контентті бөгеіп тастау үшін желіаралық экран және/немесе прокси-сервер арқылы ұйымдастыру аясында жүзеге асырылуы мүмкін. Жоғарыда талкыланып жатырған көптеген мәселелер сияқты хабардарлық.

Сандық қоқыс

Адамдар өздерінің ойларын Facebook немесе Twitter арқылы орналастыруға немесе түрлі блоктар мен хабарландыру тақталарын түзетуге үйреніп қалғаны соншалық, олар салатын ақпаратына немқұрайлы қарайды.

Келесіні қарастырып көрелік, «бүгін үйге 3 шамасында FAM-мен кино көруге барайық» байғұс жаңартуы сияқты еді.

Осы адамның аздаған мөлшерде ұрлауға ұялмайтын «досы» бар деп елестетіңіз, енді досыңыз Сіздің қашан және қанша уақытқа болмайтыныңызды біледі. Кейін үйіңіздің бос болатынын жарнамалайды? не де болса, адамбыз ғой (отбасын түстікке ертіп алып демалысқа шықпақшы, түні бойы жұмыс істейді және т.б.)

Егер сенбесеңіз, онда сіздің бейкүнә хабарламаларыңыз сізді қолайсыздықтарға ұрындыруы мүмкін, азғантай зерттеу жүргізіп көріңіз. Әлеуметтік медиа және қоғамдастық арқылы адамдар туралы ақпарат алуға арналған веб-сайттардың бірқатары бар. spokeo.com сайты бұны өте мұқият зерттейді. Осыны тексеріп, аты-жөніңізді енгізіңіз. Сіз келесі ақпараттың Сізден алынған екенін анықтайсыз:

жасыңыз, отбасы жағдайыңыз, қаржылық активтер құны, саясатыңыз бен дініңіз, қызығушылықтарыңыз/хобби, жаттығу түрі, денсаулығыңыз, отбасы мүшелерінің есімдері, үйіңіздің орналасуы, үйіңіздің

құны, үйіңіздің еденінің суреті.

Егер бұл сізді қорқыту үшін жеткіліксіз болса, онда сіз туралы осындай ақпараты бар сайттардың әлі ондағаны бар деп есептеңіз:

- mylife.com
- whozat.com
- socialpulse.com
- peekyou.com

аталғандар солардың кейбіреулері!

және біліміңіз өте маңызды. Компьютерлік техниканы тиісінше пайдалануға қатысты саясат шешімдерін де ұсынуы мүмкін. Веб-жасаушылар мен әкімшілер өздерінің веб-сайты заңға және қағидаларға сәйкес келетініне кепілдік беру үшін мәселелермен таныс болуы тиіс.

Үздіксіз білім беру

Біз «оқушыны бүкіл өмірі бойы» деген терминді пайдаланып қойдық. Бұл нені білдіреді? Фраза сіз өмір бойы оқуды жалғастырасыз дегенді білдіреді. Бұл АТ-мамандары үшін өте маңызды, өйткені АТ өзгеруін жалғастыра береді. Осы мәтінде берілген компьютерлік аппараттық құралдардың, бағдарламалық қамтаманың, операциялық жүйелердің және бағдарламалау тілінің түрлі тарихтарын еске түсірейік. Әр сала (компьютерлік инженерия, бағдарламалық инженерия, инженерлік жүйелер) осы салада жұмыс істейтін қызметкерлер оқуды жалғастыруын талап етеді. Мұнсыз, өйткені жаңа технологиялар пайда болуда, жұмыскер оларды пайдалана алмайды. Жұмыскер енді компанияны өзін жетілдіруден артқа ұстайды. Осындай жағдайларда жұмыскердің жауапкершілік арқалайтыны және ақырында оны жұмыстан шығаруы әбден мүмкін. Осылайша қызмет өрісіне бағдарланған кез-келген технологияда үздіксіз білім алу шешуші маңызға ие. АТ-мамандары үшін үздіксіз білім алу ұйымыңыз үшін құндылық мәселесі ғана емес, ұйымыңызға технологияның болашақ бағыттары туралы негізгі шешімдер қабылдауға көмек болып табылады.

АТ-да қолжетімді үздіксіз білім берудің көптеген түрлері бар. Бұл түрлердің көбі айналаңызда кезіп жүр, сондықтан Сіз оларды қалауыңызша игересіз. Оқуға қолжетімді көптеген дереккөздер бар, бірақ бұл ол дереккөздерді тауып, оларды пайдалану үшін бастамашылықты өзіңізге алу үшін. Сауда журналдарында қолжетімді форумдардың арасында олар Wired, New Scientist, доктор Dobb журналы сынды немесе IEEE және ACM жарияланымдар сынды атақты баспасөздің бөлігі болып

табыла ма. 16.2-кестеде IEEE және ACM жарияланымдарынан кейбір өзектілерінің тізімі келтірілген. *

Журналдарға көбірек бағытталған оқумен байланысты екі проблема олардың, әдеттегідей, қолжетімді салалардағы технологияларды емес, озық зерттеулерді атап көрсететінінде әрі олардың аспиранттар мен оқытушылар үшін икемдірек болатынында. Осылайша, АТ-мамандары ретінде сіз ең алуан түрлі кітаптар мен веб-сайттардың көмегімен технологияны зерттеуді жөн көресіз.

Технологиялық кітаптардың үлкен нарығы бар. Шын мәнінде, шамамен әрбір ІТ тақырыбы «жалған» кітаптар стиліне бағытталған пайдаланушылар басшылығы тарапынан шығарылған бірнеше ондаған кітаптардан тұруы мүмкін. Атақтылау АТ-шығарушылардың арасында - Томпсон Course Technologies, O'Reilly, Elsevier және Springer Verlag.

Сайттарға бағдарланған көптеген технологиялар АТ-маманы тарапынан жасалып, демеліп отырады. Олар оқи отырып, пайда әкеле алатын басқа адамдар үшін сервис ретінде өз білімдерін жариялайды. Мәселен, Linux операциялық жүйесі, Windows 7 операциялық жүйесі, Интернет, Apache веб-сервері сынд және т.б. туралы ақпараты бар сайттардың саны өте көп. Бұл сайттар сізге білім беру «ретінде» ғана емес, белгілі проблемалар бойынша заманауи ақыл-кеңестер береді және қауіпсіздіктің қыр-сырын үйретеді.

* IEEE туралы қосымша ақпарат алу үшін www.ieee.org қараңыз, сондай-ақ ACM туралы қосымша ақпарат алу үшін www.acm.org сайтына кіріңіз.

16.2-кесте ACM және IEEE ІТ-Бағдарланған Журналдар Журнал Аты АТ-қараңыз

ACM Ақпарат мәмілелері және қауіпсіздік жүйелері

Криптография, авторландыру, аудит тетіктері, хаттамалар, қауіптер, деректер бүтіндігі, жеке өмірдің қол сұғылмаушылық мәселелері.

ACM сандық тасығыштар мен сандық құқықтардың интернет-технологияларына жасалатын мәмілелер, электронды коммерция, Интернет Қылмыс пен заң, Интернет өндірімділігі, тең бір дәрежелі желілер, қауіпсіздік пен құпиялылық

ACM Сақтау және архитектура үшін сақтау жүйелері бойынша мәмілелер, сақтаулар,

желілерді, виртуалдандыруды, апатты қалпына келтіруді, кәштеуді GRID сақтау аймақтары
АСМ веб-браузер интерфейстеріне жасалатын мәмілелер, электронды коммерция, веб-сервистер, XML, қолжетімділік, өндірімділік, қауіпсіздік және құпиялылық
Коммуникациялар АСМ Коммуникациялар/желілер, компьютерлік жүйелер, компьютерлер мен қоғам, деректерді сақтау және іздеу, өндірімділік жүйелері, дербес компьютерлер, қауіпсіздік
EEE Internet Computing IT-қызметтер, WWW технологиялары
IEEE желі хаттамалары мен құрылымы, хаттаманы жобалау, байланысты, желілік басқаруды және енгізуді бағдарламалық қамтамасыз ету
IEEE технологиялары және қоғам Қоғамға әсер ету, әлеуметтік аспектілердің тарихы, кәсіби міндеттер
Операциялық жүйе компьютерлеріндегі IEEE операциялары, коммуникациялық хаттамалар, өндірімділік, қауіпсіздік

Сенімділік және Secure Computing бойынша IEEE операциялары
Сымсыз байланыс бойынша IEEE операциялары

Бағалау сенімділігі мен қауіпсіздігін операциялық жүйелер мен желілерді, модельдеу әдістерін, қауіпсіздік, өнімділігі теориялық және тәжірибелік тақырып тарсымсыз байланыс жүйелерін қоса алғанда, прототипін және жаңа қосымшалар АТ-мамандарын ұйымдастыру, деректерді кросс-функционалдық жүйелердің, АТ-жарылу, IT-жетістіктер жаңа технологиялар, электрондық коммерция, топтық жұмыс, кең жолақты желілер, қауіпсіздік құралдары.

Егер сіз өз қалауы бойынша, өте ыңғайлы оқытуды басшылыққа ала отырып, нысандарын оқуын жалғастыру әдісі бойынша қол жетімді. Олар курстарды қамтиды (ол онлайн арқылы техникалық мектептер арқылы сіздің жеке ұйымға немесе ұсынылатын университеттері), сізді толықты дайындау мүмкіндігінемесе немесе білімін тереңдетіп кеңейте алады. Көптеген компаниялар өз АТ-мамандарын төлеп, түрлі нысанда үздіксіз білім алуын қолдайтын болды.

Мысалы, ұйым ғылыми дәреже іздестіруге оқыту немесе сертификат алу үшін төлейді. АТ-Мансабындағы сертификаттардың алуантүрлілігі. Бүгінгі таңда АТ байланысты сертификаттардың өте үлкен саны бар. АТ жұмыс орындарында Сіз сұхбат алу кезінде, ұйымның бұл сертификаттарды бағалы қағаздар деп санайтындығы туралы сұрай аласыз ба?

Сондай-ақ, барлық әлемде жылына бірнеше рет болатын сауда конференциясы бар. Бұл конференцияның авторлары Adobe, Oracle, VMware, Microsoft және IBM қамтиды. Басқалары бұлтты есептеулер, веб-сервистерді есептеу сияқты тақырыптарды қамтиды. Web 2.0, деректер базасын, желілік байланыстар, виртуалды, функционалдық үйлесімділік, жаңа технологиялар, криптографиялық және АТ озық тәжірибесін олардың кейбіреулерін атау үшін ғана. Және біліктілікті арттыру курстары сияқты, өз АТ-мамандарының осындай конференцияларға қатысуын төлеу үшін ол ұйым үшін ортақ болып табылады.

Істің мәні бойынша, АТ-білім алуды жалғастыру үшін, сіз оқуға дайын болуыңыз керек. Сіз енді өзіңіз оқып жатырсыз және өзіңіздің біліміңізді жалғастырасыз, сіздің жұмыс берушіңіз оны бағалы деп тапқаннан немесе оның сізге қызықты болғандығынан емес, сіздің бұнымен ләззаттанатыныңызға сенемін.